Bidirectional Representations Augmented Autoregressive Biological Sequence Generation

Xiang Zhang[1,2,†*] **Jiaqi Wei**[3,4*] **Zijie Qiu**[1,3] **Sheng Xu**[1,3] **Zhi Jin**[3]
Zhiqiang Gao[3] **Nanqing Dong**[3] **Siqi Sun**[1,3]
[1] Fudan University [2] University of British Columbia
[3] Shanghai Artificial Intelligence Laboratory [4] Zhejiang University
xzhang23@ualberta.ca, siqisun@fudan.edu.cn

Abstract

Autoregressive (AR) models, common in sequence generation, are limited in many **biological tasks** like *de novo* peptide sequencing and protein modeling by their unidirectional nature, **failing to capture crucial global bidirectional token dependencies**. Non-Autoregressive (NAR) models offer holistic, **bidirectional** representations but face challenges with generative coherence and scalability. To transcend this, we propose a hybrid framework enhancing AR generation by dynamically integrating rich contextual information from non-autoregressive mechanisms. Our approach couples a *shared input encoder* with two decoders: a non-autoregressive one learning latent bidirectional biological features, and an AR decoder synthesizing the biological sequence by leveraging these bi-directional features. A *novel cross-decoder attention* module enables the AR decoder to iteratively query and integrate these bidirectional features, enriching its predictions. This synergy is cultivated via a tailored training strategy with *importance annealing* for balanced objectives and *cross-decoder gradient blocking* for stable, focused learning. Evaluations on a demanding 9-species benchmark of *de novo* peptide sequencing task show our model substantially surpasses AR and NAR baselines. It uniquely harmonizes AR stability with NAR contextual awareness, delivering robust, superior performance on diverse downstream data. This research advances biological sequence modeling techniques and contributes a novel architectural paradigm for augmenting AR models with enhanced bidirectional understanding for complex sequence generation. Our code is available in GitHub.

1 Introduction

Biological sequences—including DNA, RNA, and proteins—encode the fundamental information of life [35, 49]. Modeling these sequences requires capturing not only local motifs but also **global bidirectional dependencies**, as distant tokens and elements often interact in ways that critically determine biological function [1, 23, 14]. For example, amino acid residues may form structural or functional interactions across long sequence distances from both directions, and motifs in DNA/RNA frequently depend on upstream and downstream context [51]. Thus, methods limited to unidirectional modeling struggle to fully capture the biological reality where sequence meaning arises from **bidirectional contextual information**.

Autoregressive (AR) models have become a dominant paradigm in biological sequence generation due to their capacity for tractable sequential modeling [14]. However, their inherent unidirectionality imposes a fundamental constraint, limiting their ability to capture the global sequence semantics

[*] Equal contributions. † Work done while interning at Fudan University.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

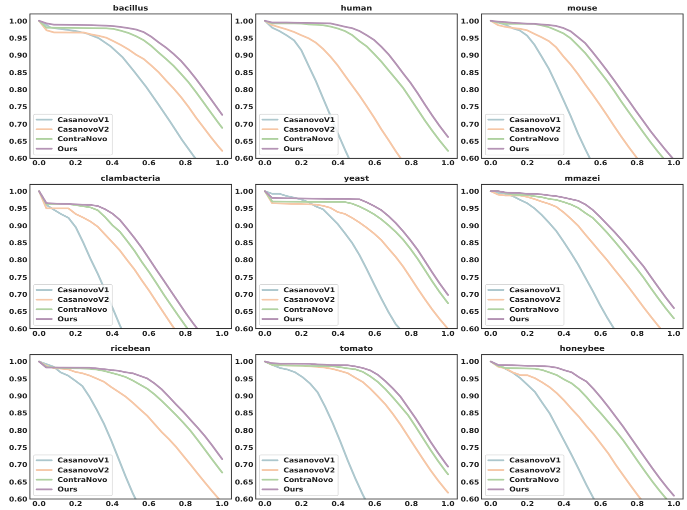

Figure 1: **Peptide Precision-Coverage Curves for Various Species**. Across all subplots, the green lines are consistently positioned above the blue and orange lines, illustrating the superior performance of our model in peptide recall over varying coverage levels. (X-axis: coverage of peptides according to confidence scores, Y-axis: Peptide recall)

often crucial for accurately interpreting biological sequences. In contrast, Non-Autoregressive (NAR) Transformers (NATs) employ holistic bidirectional self-attention mechanisms, enabling them to learn potent bidirectional representations. This characteristic allows NATs to better reflect underlying biological realities and potentially achieve superior performance [46]. Despite these representational advantages, NATs introduce practical and theoretical challenges. Notably, managing variable sequence lengths often necessitating the use of auxiliary length predictors that can be prone to errors [19], navigating complex bi-directional optimization landscapes (e.g., using NAT-specific loss [10, 11]) can lead to training instability and reduced scalability compared to their AR counterparts [26]. These inherent difficulties have, to date, hindered the full realization of NATs' bidirectional processing strengths in the domain of biological sequence modeling.

To transcend the prevailing dichotomy between the generative stability of AR models and the rich contextual understanding afforded by NAR, we introduce **CrossNovo**, a novel hybrid framework. This architecture empowers autoregressive generation by dynamically integrating potent contextual information derived from non-autoregressive mechanisms. Our approach couples a *shared encoder* for input condition encoding, with two specialized decoders for sequence decoding. The first, a non-autoregressive decoder, is tasked with learning latent bidirectional contextual features directly from the input. Concurrently, an autoregressive decoder synthesizes the sequence, its generation process critically informed by these contextual features. The cornerstone of this integration is a novel cross-decoder attention module, which enables the autoregressive decoder to iteratively query and incorporate the rich bidirectional representations captured by its non-autoregressive counterpart. This symbiotic relationship is cultivated through a training strategy incorporating *importance annealing* for balanced multi-objective optimization and *cross-decoder gradient blocking* to ensure focused representational learning and maintain training stability.

In this work, we focus on the challenging task of *de novo* peptide sequencing as a **test bed** for evaluating our framework. Peptide sequencing from tandem mass spectrometry (MS/MS) data is pivotal in proteomics [2], impacting research from fundamental biology to drug development [1, 23]. However, traditional database search methods [8, 25, 45, 6, 43] falter with novel sequences, as seen in *de novo* antibody characterization [3], neoantigen discovery [17], and metaproteomics [13]. *De novo* sequencing, inferring sequences directly from spectra, is thus indispensable. The problem is well-structured, computationally demanding, and highly consequential for protein sequence modeling, making it an ideal benchmark for exploring hybrid AR–NAR architectures.

On extensive 9-species benchmarks [31, 40, 41, 49, 42], CrossNovo substantially surpasses AT and NAT baselines by uniquely harmonizing AT generative stability with NAT contextual awareness, yielding robust, superior performance. Ablation studies validate our architectural innovations, including the cross-decoder attention. This work significantly advances biological sequence modeling and

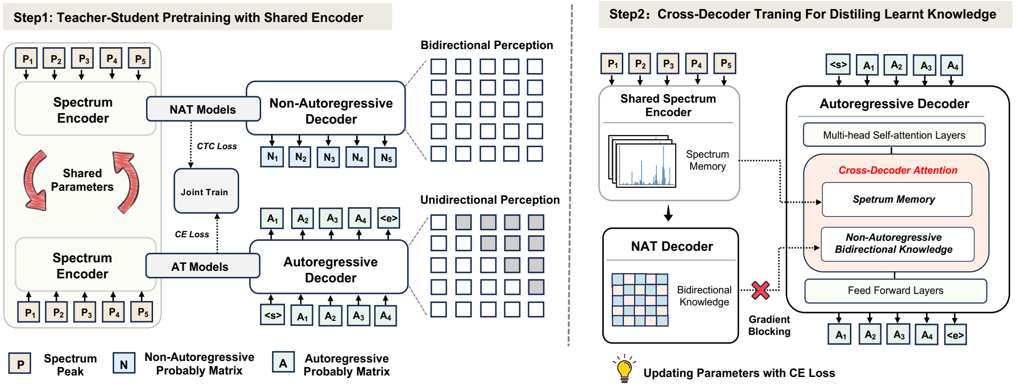

Figure 2: **The architecture of CrossNovo**. Step 1 involves joint training with a shared encoder in a multitask learning framework, enabling the simultaneous training of Autoregressive (AT) and Non-Autoregressive (NAT) decoders. This approach exploits the synergistic advantages of multitask learning to enhance performance. Upon convergence of both decoders, Step 2 introduces a novel knowledge distillation process, transferring insights from the NAT module to the AT module through a cross-decoder attention mechanism. Cross-decoder gradient blocking is employed throughout to optimize the training process.

de novo peptide sequencing, offering a novel paradigm for augmenting AT models with bidirectional understanding for complex sequence generational tasks.

2 Related Work

Autoregressive and Non-Autoregressive Transformers. Transformer architectures [32] underpin modern sequence modeling. Autoregressive (AT) variants generate tokens sequentially, ensuring high quality but suffering from inference latency [28, 42, 34]. Non-Autoregressive Transformers (NATs) [12, 21, 38] generate tokens in parallel for speed, though sometimes trading off accuracy [46, 47, 16]. In *de novo* peptide sequencing, prevalent AT models [40, 15] can suffer from error propagation. Our work pioneers a novel approach where a non-autoregressive decoder's insights into bidirectional context are used not for direct generation, but to substantially enrich and guide a primary autoregressive decoder, thereby addressing this limitation.

De Novo Peptide Sequencing. Early *de novo* methods [20, 9] have largely been superseded by deep learning. DeepNovo [31] initiated this shift using CNN-LSTMs. Transformer models now define the state-of-the-art. AT systems like Casanovo [40, 41] and its derivatives (e.g., AdaNovo [36], HelixNovo [39], InstaNovo [7], SearchNovo [37], ContraNovo [15], RankNovo [28]) have focused on architectural refinements within the AT paradigm. PrimeNovo [46], RefineNovo [48, 47], and XuanjiNovo [16] explored NAT generation for speed, though NATs can present challenges with sequence length prediction and complex optimization [19].

CrossNovo introduces a distinct and novel framework that transcends the conventional AT/NAT dichotomy. We propose a unique autoregressive architecture fundamentally enhanced by rich bidirectional latent representations. These representations are innovatively distilled from an auxiliary non-autoregressive process, allowing our model to integrate the precision of AT decoding with the holistic contextual understanding of NATs. This pioneering hybrid formulation achieves superior accuracy by uniquely leveraging the strengths of both approaches for *de novo* peptide sequencing.

3 Method

This section elucidates the architectural design and training paradigm of **CrossNovo**, a novel hybrid framework engineered for *de novo* peptide sequencing. We first establish the problem formulation and notation. Subsequently, we dissect the core architectural innovations of **CrossNovo**, comprising a shared spectral feature extractor, distinct autoregressive (AT) and non-autoregressive (NAT) peptide decoders, and crucially, the sophisticated mechanisms architected for their synergistic knowledge integration.

3.1 Problem Formulation and Notation Overview

The core challenge of *de novo* peptide sequencing is the direct inference of an amino acid sequence $\mathbf{A} = (\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_n)$ from an observed mass spectrum $\mathcal{S}$. We formally define the input $\mathcal{S}$ as a composite of three critical information sources: (i) the peak list $\mathcal{I} = \{(\mathrm{mz}_j, \mathrm{g}_j)\}_{j=1}^{k}$, representing detected mass-to-charge ratios (mz) and their respective intensities (g); (ii) the precursor mass $\mathbf{m} \in \mathbb{R}^+$; and (iii) the precursor charge state $\mathbf{c} \in \mathbb{Z}^+$. Our objective is to learn a mapping $f : \mathcal{S} \mapsto \mathbf{A}$ that accurately predicts the ground-truth sequence.

3.2 Model Backbone of CrossNovo

The architectural blueprint of **CrossNovo** centers on a novel tripartite structure: a foundational shared spectrum encoder coupled with two specialized decoders—an autoregressive (AT) head for high-fidelity sequential generation and a non-autoregressive (NAT) head engineered to capture global sequence context. This dual-decoder system is designed to synergistically harness the complementary strengths of autoregressive precision and non-autoregressive holistic understanding.

Shared Spectrum Encoder. The core of **CrossNovo**'s representation learning lies in its shared spectrum encoder, a Transformer-based architecture tasked with transforming raw spectral data $\mathcal{S}$ into a rich, latent representation. We conceptualize the input peak list $\mathcal{I}$ as a sequence of tokens. For each peak $(\mathrm{mz}_i, \mathrm{g}_i)$, its constituent mass-to-charge ratio mz_i and intensity g_i are independently projected into d-dimensional embeddings using a domain-adapted sinusoidal encoding.

$$\mathbf{e}_j^0(\mathbf{v}) = \begin{cases} \sin((\mathbf{v})/(C \cdot (\frac{(\mathbf{v})_{\min}}{2\pi})^{\frac{2j}{d}})), & \text{for } j \le \frac{d}{2} \\ \cos((\mathbf{v})/(C \cdot (\frac{(\mathbf{v})_{\min}}{2\pi})^{\frac{2j}{d}})), & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{v}$ is the float value (either mz_i or g_i) to be encoded, $C = (\mathbf{v})_{\max}/(\mathbf{v})_{\min}$ is a scaling factor, with $(\mathbf{v})_{\max}$ and $(\mathbf{v})_{\min}$ being the pre-defined bounds for the values. These per-value embeddings are subsequently fused, typically via summation, to yield an initial peak embedding sequence $\mathbf{E}^{(0)} = (\mathbf{e}_1^{(0)}, \ldots, \mathbf{e}_k^{(0)})$. This sequence is then refined through a stack of b canonical self-attention layers:

$$\mathbf{E}^{(i)} = \text{SelfAttentionLayer}(\mathbf{E}^{(i-1)}) \quad \text{for } i = 1, \ldots, b. \tag{2}$$

The resultant high-level feature sequence $\mathbf{E}^{(b)} = (\mathbf{e}_1^{(b)}, \mathbf{e}_2^{(b)}, \ldots, \mathbf{e}_k^{(b)})$ constitutes the pivotal shared spectral context, concurrently utilized by both the AT and NAT decoders.

AT Peptide Decoder. For sequential peptide construction, **CrossNovo** incorporates an autoregressive (AT) decoder, which instantiates a Transformer decoder architecture carefully tailored for the nuances of peptide sequence generation. Conditioned on previously generated amino acids (or ground-truth tokens during teacher forcing), represented as embeddings $\mathbf{H}^{(0)}$, the AT decoder iteratively predicts the next amino acid a_t. This process unfolds over L decoder layers, where each layer performs causal self-attention, producing $\mathbf{h}_t'^{(l)} = \text{CausalSelfAttn}(\mathbf{h}_t^{(l-1)}, \{\mathbf{h}_{1:t-1}^{(l-1)}\})$, to maintain autoregressive integrity. This is followed by cross-attention to the shared spectrum features $\mathbf{E}^{(b)}$, yielding $\mathbf{h}_t''^{(l)} = \text{CrossAttn}(\mathbf{h}_t'^{(l)}, \mathbf{E}^{(b)})$, and finally a feed-forward network. The output representation $\mathbf{h}_t^{(L)}$ for each token t from the terminal layer sequence $\mathbf{H}^{(L)}$ is then mapped to a probability distribution over the amino acid vocabulary using $P_t(\cdot \mid \mathcal{S}, \mathrm{a}_{<t}) = \text{softmax}(\boldsymbol{\mathcal{W}}\mathbf{h}_t^{(L)})$. The token a_t is then sampled from this distribution. To further bolster its predictive accuracy, and drawing inspiration from established biochemical constraints [15], our AT decoder is uniquely augmented with prefix and suffix mass information at each generation step, thereby injecting vital biological context directly into the decoding process.

NAT Peptide Decoder. Complementing the AT decoder, **CrossNovo** integrates a non-autoregressive (NAT) decoder specifically engineered to learn a holistic, global understanding of the peptide sequence from the spectrum. Diverging from the sequential nature of the AT component, the NAT decoder generates the entire peptide sequence in a single pass. Architecturally, it mirrors a Transformer decoder but strategically omits the causal mask in its self-attention mechanisms, computing $\mathbf{v}_t'^{(l)} = \text{SelfAttn}(\mathbf{v}_t^{(l-1)}, \mathbf{V}^{(l-1)})$ to enable true bidirectional context aggregation. Adopting a common strategy for NAT models [46], we operate with a predefined maximum sequence length $T_{\max}$. A pivotal design choice for fostering effective knowledge distillation to the AT decoder (detailed in Section 3.4) is the NAT decoder's input: it exclusively receives positional embeddings $\mathbf{P}^{(0)} = (\mathbf{p}_1, \ldots, \mathbf{p}_{T_{\max}})$, without any target sequence token information. These embeddings are processed through L' layers, each comprising this non-causal self-attention and subsequent cross-attention to the shared spectral features $\mathbf{E}^{(b)}$, given by $\mathbf{v}_t''^{(l)} = \text{CrossAttn}(\mathbf{v}_t'^{(l)}, \mathbf{E}^{(b)})$. The resulting latent representations $\mathbf{V}^{(L')}$ from the final NAT layer are then linearly transformed to yield predictions for all $T_{\max}$ positions concurrently.

3.3 Novel Joint Training Strategy for CrossNovo

A central contribution of **CrossNovo** lies in its sophisticated joint training strategy, meticulously designed to concurrently optimize both the AT and NAT decoders while fostering a unique unidirectional knowledge flow from the NAT to the AT component. This co-training paradigm leverages a shared encoder and optimizes the entire architecture via a carefully constructed multitask loss function.

Multitask Learning Loss. The combined loss function $\mathcal{L}_{\texttt{total}}$ is a weighted sum of the AT and NAT decoder losses:

$$\mathcal{L}_{\texttt{total}} = \lambda_{\texttt{AT}}\mathcal{L}_{\texttt{AT}} + (1 - \lambda_{\texttt{AT}})\mathcal{L}_{\texttt{NAT}} \tag{3}$$

The autoregressive loss $\mathcal{L}_{\texttt{AT}}$ is the standard cross-entropy loss:

$$\mathcal{L}_{\texttt{AT}} = -\log P(\mathbf{A} \mid \mathcal{S}; \theta) = -\sum_{t=1}^{n} \log p(\mathrm{a}_t \mid \mathrm{a}_{<t}, \mathcal{S}; \theta) \tag{4}$$

where θ represents the model parameters.

For the NAT decoder, we employ the Connectionist Temporal Classification (CTC) loss [10] to effectively handle variable-length sequences and learn robust bidirectional representations. Given a maximum generation length $T_{\max}$, the NAT decoder produces a sequence $\mathbf{y} = (\mathrm{y}_1, \ldots, \mathrm{y}_{T_{\max}})$. The CTC mechanism introduces a blank token ϵ and defines a reduction function Γ that removes repeated tokens and then blank tokens (e.g., $\Gamma(\text{AAT}\epsilon\text{TG}) = \text{ATTG}$). The NAT loss aims to maximize the sum of probabilities of all paths $\mathbf{y}$ that can be reduced to the target sequence $\mathbf{A}$:

$$\mathcal{L}_{\texttt{NAT}} = -\log \left(\sum_{\mathbf{y}:\Gamma(\mathbf{y})=\mathbf{A}} \prod_{j=1}^{T_{\max}} P(\mathrm{y}_j | \mathcal{S}, \theta) \right) \tag{5}$$

***Importance Annealing* for Balanced Optimization.** Recognizing that the NAT decoder's strength lies in learning rich contextual priors while the AT decoder excels at fine-grained sequential prediction, we introduce a scheduling mechanism termed *importance annealing* for the weighting coefficient $\lambda_{\texttt{AT}}$. This technique, drawing conceptual parallels with curriculum learning strategies, dynamically modulates the relative contributions of the AT and NAT losses throughout training. Here, the weighting coefficient $\lambda_{\texttt{AT}}(i)$ progressively increases with training iteration i according to the relation $\lambda_{\texttt{AT}}(i) = i/T_{\text{total}}$, where i is the current training iteration and T_{total} is the total number of iterations. This strategic annealing ensures that the NAT decoder initially plays a dominant role in shaping robust spectral representations, with the optimization landscape gradually transitioning to prioritize the AT decoder's objective for refining precise, high-fidelity sequence generation in later stages.

3.4 Novel Cross-Decoder Attention for Knowledge Transfer

The principal innovation enabling synergistic AT-NAT interaction within **CrossNovo** is our proposed *cross-decoder attention* mechanism. This novel module facilitates the direct injection of rich, bidirectionally-informed contextual knowledge, pre-learned by the NAT decoder, into the AT decoder's generation process. This transfer is typically activated during a dedicated second fine-tuning stage or at inference. Specifically, we re-engineer the AT decoder's cross-attention submodule:

Algorithm 1 **CrossNovo**: AT Fine-tuning with Cross-Decoder NAT Knowledge Transfer

```
Inputs: Dataset $\mathcal{D}$; Pre-trained parameters $(\theta_{\text{enc}}, \theta_{\text{AT}}, \theta_{\text{NAT}})$ from Stage 1; Fine-tuning epochs $E_{\text{ft}}$; Learning rate $\eta_{\text{ft}}$; Max NAT length $T_{\max}$.
if $E_{\text{ft}} > 0$ then
    Freeze parameters $\theta_{\text{enc}}$ and $\theta_{\text{NAT}}$.
    for epoch $e = 1$ to $E_{\text{ft}}$ do
        for each $(\mathcal{S}, \mathbf{A})$ in $\mathcal{D}$ do
            $\mathbf{E}^{(b)} \leftarrow \text{Encoder}(\mathcal{S}; \theta_{\text{enc}})$    ▷ Use frozen encoder
            $\mathbf{V}^{(L')} \leftarrow \text{NATForward}(\text{PositionalEmbeddings}(T_{\max}), \mathbf{E}^{(b)}; \theta_{\text{NAT}})$    ▷ NAT features from frozen NAT decoder
            $\mathbf{V}^{(L')}_{\text{blocked}} \leftarrow \mathbb{GB}(\mathbf{V}^{(L')})$    ▷ Cross-decoder gradient blocking
            $\mathbf{C}_{\text{aug}} \leftarrow [\mathbf{V}^{(L')}_{\text{blocked}} \oplus \mathbf{E}^{(b)}]$    ▷ Augmented context for AT, cf. Eq. 8, 10 (with distinct positional encodings)
            $\mathcal{L}_{\texttt{AT-ft}} \leftarrow \text{ComputeATLossAugmented}(\mathbf{A}, \mathbf{C}_{\text{aug}}; \theta_{\text{AT}})$
            Update $\theta_{\text{AT}}$ using $\nabla_{\theta_{\text{AT}}} \mathcal{L}_{\texttt{AT-ft}}$ with $\eta_{\text{ft}}$.
Return Fine-tuned parameters $\theta_{\text{AT}}$ (and unchanged $\theta_{\text{enc}}, \theta_{\text{NAT}}$).
```

instead of solely attending to the primary spectrum features $\mathbf{E}^{(b)}$, the AT decoder's query states $\mathbf{h}_t^{\prime(l)}$ now attend to an augmented context. This augmented context is formed by concatenating the NAT decoder's final layer latent representations $\mathbf{V}^{(L')}_{\text{p}\{1:T_{\max}\}}$ with the original spectrum features $\mathbf{E}^{(b)}_{\text{p}\{T_{\max}+1:T_{\max}+k\}}$, as shown in Equation 6.

$$\mathbf{h}_t^{\text{update}} = \text{CrossAttn}\left(\mathbf{h}_t^{\prime(l)}, \left[\mathbf{V}^{(L')}_{\text{p}\{1:T_{\max}\}} \oplus \mathbf{E}^{(b)}_{\text{p}\{T_{\max}+1:T_{\max}+k\}}\right]\right) \tag{6}$$

Here, $\oplus$ denotes concatenation along the sequence dimension. The distinct positional encodings (denoted by subscript $\text{p}\{\cdot\}$) ensure the AT decoder can differentiate and appropriately leverage these heterogeneous information sources. Optimization during this knowledge transfer phase is driven exclusively by the AT loss, $\mathcal{L}_{\texttt{AT}}$.

Cross-Decoder Gradient Blocking **for Stable Learning.** A critical consideration when fusing representations from independently (or jointly but distinctly) optimized modules is the potential for undesirable gradient interference. If the AT decoder's loss $\mathcal{L}_{\texttt{AT}}$ were allowed to backpropagate through the NAT-derived features $\mathbf{V}^{(L')}_{\text{p}\{1:T_{\max}\}}$, it could perturb the carefully learned representations optimized under the NAT-specific CTC loss ($\mathcal{L}_{\texttt{NAT}}$), a phenomenon we term 'representational drift.' To preempt this, we institute a *cross-decoder gradient blocking* strategy. As formalized in Equation 7, the NAT features $\mathbf{V}^{(L')}_{\text{p}\{1:T_{\max}\}}$ are treated as immutable, pre-computed contextual inputs by the AT decoder, effectively detaching them from the AT decoder's computation graph with respect to gradient flow back into the NAT decoder (e.g., via 'torch.Tensor.detach()').

$$\text{h}_t^{\text{update}} = \text{CrossAttn}\left(\text{h}_t^{\prime(l)}, \left[\mathbb{GB}\left(\mathbf{V}^{(L')}_{\text{p}\{1:T_{\max}\}}\right) \oplus \mathbf{E}^{(b)}_{\text{p}\{T_{\max}+1:T_{\max}+k\}}\right]\right) \tag{7}$$

where $\mathbb{GB}(\cdot)$ denotes the gradient blocking operation. This isolation ensures that the AT decoder can effectively assimilate the distilled knowledge from the NAT branch without corrupting its source, thereby promoting stable training dynamics and enhancing the performance of **CrossNovo**.

3.5 Consideration of Bidirectional Knowledge Transfer

While our proposed knowledge transfer is unidirectional (NAT→AT), one might contemplate a reciprocal AT→NAT pathway. Although architecturally plausible—for instance, by enabling the NAT decoder to attend to AT hidden states—such a design introduces a substantial risk of data contamination. Specifically, due to the teacher-forcing regimen commonly employed during AT decoder training, its internal representations are directly exposed to ground-truth target tokens. Allowing the NAT decoder to access these states would inadvertently provide it with privileged information, circumventing the fundamental challenge of *de novo* prediction from spectral data alone. Our NAT→AT transfer mechanism, in contrast, is intrinsically robust against such leakage.

The NAT decoder, by its design of solely using positional inputs for its sequence generation task (before its features are passed), learns representations purely from the spectrum and its own output distribution's constraints. Consequently, the features it provides to the AT decoder are untainted by ground-truth sequence information from the AT, ensuring the integrity of the training process. This carefully designed unidirectional distillation allows **CrossNovo** to effectively leverage the NAT's bidirectional context while rigorously upholding the principles of fair and challenging model training.

4 Experiments

4.1 Experiments Setup

Dataset. Following prior work [41, 46] for fair comparison, we trained **CrossNovo** on the MassIVE-KB dataset [33], which contains 30 million high-resolution peptide-spectrum matches (PSMs) from diverse instruments. For validation and benchmarking against leading methods [40, 41, 46, 50], we used the 9-species-v1 (approx. 1.5M spectra from nine experiments) and the 9-species-v2 revised datasets. The latter offers more and higher-quality spectra with broader data distribution and stricter annotation than its predecessor.

Implementation Details. All inputs (peaks and amino acids) were embedded into 400 dimensions. The shared spectrum encoder, NAT decoder, and AT decoder of **CrossNovo** each comprise 9 Transformer layers with 8 attention heads and 1024 hidden dimensions. We trained **CrossNovo** on eight NVIDIA A100 80GB GPUs using the AdamW optimizer [18] with an initial learning rate of 5×10^{-4}, a linear warm-up phase, and a subsequent cosine decay schedule for training stability.

Evaluation Metrics. Following standard practice, we evaluated performance at amino acid ($\texttt{AA}$) and peptide levels. **AA-level accuracy**: An amino acid is considered correctly predicted if its mass deviation from the actual amino acid is less than 0.1 Da, and its prefix or suffix mass differences do not exceed 0.5 Da relative to the corresponding segment of the ground truth peptide. Accuracy is then $\frac{\mathbf{M}_{\texttt{AA}}}{\mathbf{T}_{\texttt{AA}}}$, where $\mathbf{M}_{\texttt{AA}}$ is the count of correctly predicted AAs and $\mathbf{T}_{\texttt{AA}}$ is the total number of predicted AAs. **Peptide-level precision**: A peptide is considered accurately predicted if all its constituent amino acids match their true counterparts. Precision is $\frac{\mathbf{M}_{\text{pep}}}{\mathbf{T}_{\text{pep}}}$, where $\mathbf{M}_{\text{pep}}$ is the count of correctly predicted peptides and $\mathbf{T}_{\text{pep}}$ is the total number of peptides evaluated in the dataset.

Baselines. We benchmarked **CrossNovo** against a diverse set of baselines spanning three major methodological paradigms: **Database (DB)**: Represented by Peaks [20], which performs spectral matching against a reference protein database, often using enzymatic digestion constraints (e.g., tryptic peptides) to refine the search. **Autoregressive (AT)**: This dominant paradigm includes early deep learning models like DeepNovo [31] and PointNovo [27]. More recent state-of-the-art Transformer-based AR models include Casanovo [40], HelixNovo [39], InstaNovo [7], CasanovoV2 [41] (incorporating beam search), and ContraNovo [15] (utilizing contrastive learning and amino acid mass embeddings). **Non-Autoregressive (NAT)**: A newer paradigm exemplified by PrimeNovo [46]. As the first NAR-based model for peptide generation, PrimeNovo demonstrated competitive accuracy with superior inference efficiency, highlighting the potential of NAR strategies, which **CrossNovo**'s hybrid architecture distinctively builds upon.

4.2 Results

Performance on 9-Species-v1 Benchmark Dataset. Table 1 highlights the superior performance of CrossNovo, establishing new state-of-the-art results at both amino acid and peptide levels. Our model outperforms prior autoregressive (AR) methods across all species and metrics, with amino acid recall improving from 0.785 to 0.811 and peptide recall from 0.621 to 0.654. The recall-coverage graph in Figure 1 further illustrates its dominance, consistently outperforming all baseline models at every coverage level. Moreover, our knowledge distillation techniques successfully bridge the gap between AT and NAT. Our model not only exceeds NAT in amino acid precision across all species but also surpasses NAT in peptide recall for all but two species, where it remains highly competitive. Furthermore, we observe that our combined training module has granted CrossNovo the advantages of both AT and NAT in predicting peptides of different species. Specifically, in Human and Mouse, where AT models performed significantly better than NAT models, CrossNovo extends this advantage by further outperforming NAT by 9%. In other species where NAT models performed

Table 1: Comparison of the performance of CrossNovo and baseline methods on the 9-species-v1 test set.

Metrics	Architect	Methods	*Mouse*	*Human*	*Yeast*	*M.mazei*	*Honeybee*	*Tomato*	*Rice bean*	*Bacillus*	*C. bacteria*	Average
Amino Acid Precision	DB	Peaks	0.600	0.639	0.748	0.673	0.633	0.728	0.644	0.719	0.586	0.663
	NAT	Prime.	0.784	0.729	0.802	0.801	0.763	0.815	0.822	0.846	0.734	0.788
	AT	Deep.	0.623	0.610	0.750	0.694	0.630	0.731	0.679	0.742	0.602	0.673
		Point.	0.626	0.606	0.779	0.712	0.644	0.733	0.730	0.768	0.589	0.687
		Casa.	0.689	0.586	0.684	0.679	0.629	0.721	0.668	0.749	0.603	0.667
		Insta.	0.703	0.636	0.691	0.712	0.660	0.732	0.711	0.739	0.619	0.689
		Casa.V2	0.760	0.676	0.752	0.755	0.706	0.785	0.748	0.790	0.681	0.739
		Helix.	0.765	0.665	0.768	0.784	0.757	0.721	0.793	0.816	0.681	0.750
		Contra.	0.798	0.771	0.797	0.799	0.745	0.810	0.807	0.828	0.711	0.785
		Ours	**0.816**	**0.800**	**0.814**	**0.826**	**0.785**	**0.830**	**0.831**	**0.856**	**0.740**	**0.811**
Peptide Recall	DB	Peaks	0.197	0.277	0.428	0.356	0.287	0.403	0.362	0.387	0.203	0.322
	NAT	Prime.	0.567	0.574	0.697	0.650	0.603	**0.697**	0.702	0.721	**0.531**	0.638
	AT	Deep	0.286	0.293	0.462	0.422	0.330	0.454	0.436	0.449	0.253	0.376
		Point.	0.355	0.351	0.534	0.478	0.396	0.513	0.511	0.518	0.298	0.439
		Casa.	0.426	0.341	0.490	0.478	0.406	0.521	0.506	0.537	0.330	0.448
		Helix.	0.483	0.392	0.568	0.560	0.473	0.560	0.623	0.596	0.388	0.517
		Insta	0.471	0.455	0.559	0.528	0.466	**0.732**	0.564	0.576	0.416	0.530
		Casa.V2	0.483	0.446	0.599	0.557	0.493	0.618	0.589	0.622	0.446	0.539
		Contra.	0.567	0.622	0.674	0.630	0.576	0.672	0.677	0.688	0.486	0.621
		Ours	**0.596**	**0.661**	**0.698**	**0.660**	**0.610**	0.695	**0.716**	**0.726**	0.518	**0.654**

Table 2: Comparison of the performance of CrossNovo and baseline methods on 9-species-v2 test set. The bold font indicates the best performance.

Metrics	Architect	Methods	*Mouse*	*Human*	*Yeast*	*M.mazei*	*Honeybee*	*Tomato*	*Rice bean*	*Bacillus*	*C.bacteria*	Average
Amino Acid Precision	NAT	Prime.	0.839	0.893	0.932	0.908	0.862	0.909	0.931	0.921	0.827	0.891
	AT	Casa.V2	0.813	0.872	0.915	0.877	0.823	0.891	0.891	0.888	0.791	0.862
		Contra.	0.839	0.920	0.919	0.896	0.848	0.898	0.913	0.901	0.807	0.882
		Ours	**0.857**	**0.937**	**0.939**	**0.920**	**0.880**	**0.914**	**0.939**	**0.927**	**0.837**	**0.906**
Peptide Recall	NAT	Prime.	0.627	0.795	0.884	0.812	0.742	**0.824**	0.837	0.849	**0.626**	0.777
	AT	Casa.V2	0.555	0.712	0.837	0.754	0.669	0.783	0.772	0.793	0.558	0.714
		Contra.	0.616	0.820	0.854	0.780	0.711	0.794	0.799	0.815	0.575	0.752
		Ours	**0.651**	**0.850**	**0.885**	**0.819**	**0.751**	0.816	**0.847**	**0.850**	0.607	**0.786**

better than AT models, CrossNovo leverages NAT strengths to increase its prediction accuracy by 1-3%.

Overall, our model's ability to integrate the strengths of both AR and NAR paradigms makes it a robust and adaptable solution. This dual capability ensures its effectiveness across diverse species, making it a valuable tool for wide-ranging biological applications.

Performance on 9-Species-v2 Benchmark Dataset. We evaluated **CrossNovo** on the 9-Species-v2 benchmark, known for its diverse modifications and high-quality spectra. As detailed in Table 2, **CrossNovo** achieves state-of-the-art performance, securing the highest average amino acid precision (0.906) and peptide recall (0.786) across all nine species. This demonstrates that **CrossNovo**'s architecture, which integrates bidirectional context into its autoregressive framework, yields substantial improvements over existing methods.

Performance of Amino Acids with Similar Masses. Accurately differentiating between amino acids with very similar masses is crucial for achieving precise outcomes in peptide sequencing. We conducted rigorous evaluations to assess the performance of CrossNovo on correctly predicting these amino acids. Specifically, our objective was to determine the efficacy of CrossNovo in distinguishing these challenging cases. For a comprehensive comparison, we also evaluated the performance of all AT models. The results, depicted in Figure 3, demonstrate CrossNovo's exceptional performance. The consistently better performance in identifying these easily mistaken amino acids further showcases the accuracy of CrossNovo in a more fine-grained level.

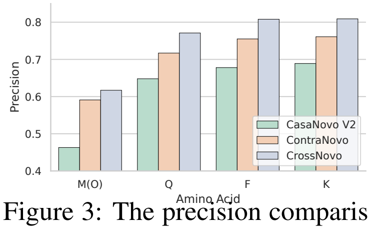

Figure 3: The precision comparison of CrossNovo with all AT models on amino acids with similar masses.

Sensitivity to Beam Size. We analyzed the effect of beam size on **CrossNovo**'s performance using the 9-species-v1 benchmark. Table 3 shows that while larger beam sizes initially improve AA Precision and Peptide Recall, these benefits plateau, and recall can slightly decrease with very large beams, possibly due to exposure bias [29]. A beam size of 5 offers an optimal trade-off between prediction accuracy and computational cost. Detailed results are in the Appendix (Section "Influence of Various Beam Sizes").

Table 3: Effect of different beam sizes on CrossNovo.

Metric	Beam Size					
	1	3	5	7	9	11
AA Precision	0.784	0.804	**0.811**	0.810	0.810	**0.811**
Peptide Recall	0.634	0.651	**0.654**	**0.654**	**0.654**	0.653

Ablation Study. The ablation study in Table 4 evaluates the effects of different proposed modules on performance. CrossNovo achieves its highest precision scores when both the cross decoder with gradient blocking and the shared encoder are utilized. In contrast, omitting the shared encoder while retaining the cross decoder and gradient blocking significantly reduces precision. Additionally, the absence of gradient blocking leads to training failure due to gradient explosion, as indicated by '✗'. These findings underscore the essential role of the cross-decoder with gradient blocking and the shared encoder in stabilizing and enhancing model performance.

Table 4: Results of the ablation study showing the effects of different model configurations. The '✗' indicates training failure due to gradient explosion.

Gradient Blocking	Cross Decoder	Shared Encoder	Amino acid Precision	Peptide Precision
		✓	0.795	0.643
	✓	✓	✗	✗
✓	✓		0.698	0.546
✓	✓	✓	**0.811**	**0.654**

Downstream Task. To demonstrate the generalizability of the proposed algorithm and its applicability, We further apply CrossNovo to a downstream task of identifying peptides in animal antibody data [4]. Obtaining the sequence information of antibodies is crucial for understanding the structural basis of antibody-antigen binding, recognition, and interaction [5]. However, existing methods for antibody protein sequencing rely on mRNA extraction from hybridoma cells [24], which can be challenging. De novo peptide sequencing offers a more accurate alternative by predicting peptides using MS/MS.

We utilize a publicly available human antibody dataset [30], which includes the light chain (LC) and heavy chain (HC) antibody proteins, each digested into peptides using various enzymes. We apply CrossNovo and several state-of-the-art baseline models to evaluate this dataset. None of these models were trained on antibody data, and all perform purely zero-shot inference. As shown in Figure 4, CrossNovo significantly outperforms the baseline models in human antibody sequencing, achieving up to a 5% improvement in both peptide recall and AA precision (Appendix Table 7). We also analyzed performance on mouse antibodies, with results detailed in the Appendix.

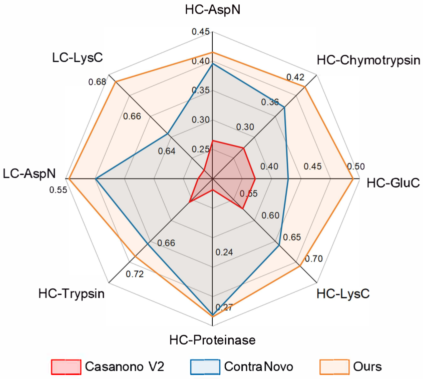

Figure 4: The peptide recall comparison of all models on Human antibody data.

5 Conclusion

In conclusion, our research presents CrossNovo, a novel approach that significantly advances de novo peptide sequencing. By effectively integrating bidirectional latent knowledge from NAT to AT, CrossNovo leverages the strengths of both AT and NAT models. Through innovative architectural modifications, CrossNovo demonstrates superior performance across diverse species, surpassing both AT and NAT baselines.

Acknowledgement

This project was fully supported by the Shanghai Artificial Intelligence Laboratory (S.S.).

References

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

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

[3] Denis Beslic, Georg Tscheuschner, Bernhard Y Renard, Michael G Weller, and Thilo Muth. Comprehensive evaluation of peptide de novo sequencing tools for monoclonal antibody assembly. Briefings in Bioinformatics, 5:1–12, dec 2022.

[4] Denis Beslic, Georg Tscheuschner, Bernhard Y Renard, Michael G Weller, and Thilo Muth. Comprehensive evaluation of peptide de novo sequencing tools for monoclonal antibody assembly. Briefings in Bioinformatics, 24(1):bbac542, 2023.

[5] Mark L Chiu, Dennis R Goulet, Alexey Teplyakov, and Gary L Gilliland. Antibody structure and function: the basis for engineering therapeutics. Antibodies, 8(4):55, 2019.

[6] Jürgen Cox and Matthias Mann. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. Nature Biotechnology, 26(12):1367–1372, 2008.

[7] Kevin Eloff, Konstantinos Kalogeropoulos, Oliver Morell, Amandla Mabona, Jakob Berg Jespersen, Wesley Williams, Sam PB van Beljouw, Marcin Skwark, Andreas Hougaard Laustsen, Stan JJ Brouns, et al. De novo peptide sequencing with instanovo: Accurate, database-free peptide identification for large scale proteomics experiments. bioRxiv, pages 2023–08, 2023.

[8] J K Eng, A L McCormack, and J R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. Journal of the American Society for Mass Spectrometry, 5(11):976–989, nov 1994.

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

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning, pages 369–376, 2006.

[11] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In International conference on machine learning, pages 1764–1772. PMLR, 2014.

[12] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. arXiv preprint arXiv:1711.02281, 2017.

[13] Robert L Hettich, Chongle Pan, Karuna Chourey, and Richard J Giannone. Metaproteomics: harnessing the power of high performance mass spectrometry to identify the suite of proteins that control metabolic activities in microbial communities. Analytical chemistry, 85(9):4203–4214, may 2013.

[14] Ming Hu, Chenglong Ma, Wei Li, Wanghan Xu, Jiamin Wu, Jucheng Hu, Tianbin Li, Guohang Zhuang, Jiaqi Liu, Yingzhou Lu, et al. A survey of scientific large language models: From data foundations to agent frontiers. arXiv preprint arXiv:2508.21148, 2025.

[15] 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.

[16] A Jun, Xiang Zhang, Xiaofan Zhang, Jiaqi Wei, Te Zhang, Yamin Deng, Pu Liu, Zongxiang Nie, Yi Chen, Nanqing Dong, et al. Massnet: billion-scale ai-friendly mass spectral corpus enables robust de novo peptide sequencing. bioRxiv, pages 2025–06, 2025.

[17] 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 : MCP, 18(12):2478–2491, dec 2019.

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

[19] Puyuan Liu, Xiang Zhang, and Lili Mou. A character-level length-control algorithm for non-autoregressive sentence summarization. Advances in Neural Information Processing Systems, 35:29101–29112, 2022.

[20] 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.

[21] Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. Flowseq: Non-autoregressive conditional sequence generation with generative flow. arXiv preprint arXiv:1909.02480, 2019.

[22] Clara Meister, Tim Vieira, and Ryan Cotterell. If beam search is the answer, what was the question? arXiv preprint arXiv:2010.02650, 2020.

[23] Cheuk Chi A Ng, Yin Zhou, and Zhong-Ping Yao. Algorithms for de-novo sequencing of peptides by tandem mass spectrometry: A review. Analytica Chimica Acta, page 341330, 2023.

[24] Weiwei Peng, Matti F Pronker, and Joost Snijder. Mass spectrometry-based de novo sequencing of monoclonal antibodies using multiple proteases and a dual fragmentation scheme. Journal of Proteome Research, 20(7):3559–3566, 2021.

[25] D N Perkins, D J Pappin, D M Creasy, and J S Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis, 20, 1999.

[26] Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. Glancing transformer for non-autoregressive neural machine translation. arXiv preprint arXiv:2008.07905, 2020.

[27] 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.

[28] Zijie Qiu, Jiaqi Wei, Xiang Zhang, Sheng Xu, Kai Zou, Zhi Jin, Zhiqiang Gao, Nanqing Dong, and Siqi Sun. Universal biological sequence reranking for improved de novo peptide sequencing. arXiv preprint arXiv:2505.17552, 2025.

[29] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732, 2015.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Jiaqi Wei, Bin Jiang, and Yanxia Zhang. Identification of blue horizontal branch stars with multimodal fusion. Publications of the Astronomical Society of the Pacific, 135(1050):084501, 2023.

[35] Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, et al. From ai for science to agentic science: A survey on autonomous scientific discovery. arXiv preprint arXiv:2508.14111, 2025.

[36] Jun Xia, Shaorong Chen, Jingbo Zhou, Tianze Ling, Wenjie Du, Sizhe Liu, and Stan Z Li. Adanovo: Adaptive\emph {De Novo} peptide sequencing with conditional mutual information. arXiv preprint arXiv:2403.07013, 2024.

[37] Jun Xia, Sizhe Liu, Jingbo Zhou, Shaorong Chen, Zicheng Liu, Yue Liu, Stan Z Li, et al. Bridging the gap between database search and\emph {De Novo} peptide sequencing with searchnovo. In NeurIPS 2024 Workshop on AI for New Drug Modalities, 2025.

[38] Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. A survey on non-autoregressive generation for neural machine translation and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(10):11407–11427, 2023.

[39] 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.

[40] 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.

[41] 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. BioRxiv, pages 2023–01, 2023.

[42] Chenyu You, Haocheng Dai, Yifei Min, Jasjeet S Sekhon, Sarang Joshi, and James S Duncan. Uncovering memorization effect in the presence of spurious correlations. Nature Communications, 16(1):5424, 2025.

[43] Jing Zhang, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng Zhang, Gilles A Lajoie, and Bin Ma. PEAKS DB: de novo sequencing assisted database search for sensitive and accurate peptide identification. Molecular & cellular proteomics : MCP, 11(4):M111.010587, apr 2012.

[44] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. arXiv preprint arXiv:1906.02448, 2019.

[45] Xiang Zhang, Juntai Cao, Jiaqi Wei, Chenyu You, and Dujian Ding. Why prompt design matters and works: A complexity analysis of prompt search space in llms. arXiv preprint arXiv:2503.10084, 2025.

[46] 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.

[47] Xiang Zhang, Jiaqi Wei, Zijie Qiu, Sheng Xu, Nanqing Dong, Zhiqiang Gao, and Siqi Sun. Curriculum learning for biological sequence prediction: The case of de novo peptide sequencing. arXiv preprint arXiv:2506.13485, 2025.

[48] Xiang Zhang, Jiaqi Wei, Zijie Qiu, Sheng Xu, Nanqing Dong, ZhiQiang Gao, and Siqi Sun. Curriculum learning for biological sequence prediction: The case of de novo peptide sequencing. In Forty-second International Conference on Machine Learning, 2025.

[49] Jingbo Zhou, Shaorong Chen, Jun Xia, Sizhe Sizhe Liu, Tianze Ling, Wenjie Du, Yue Liu, Jianwei Yin, and Stan Z Li. Novobench: Benchmarking deep learning-based\emph {De Novo} sequencing methods in proteomics. Advances in Neural Information Processing Systems, 37:104776–104791, 2024.

[50] Xie-Xuan Zhou, Wen-Feng Zeng, Hao Chi, Chunjie Luo, Chao Liu, Jianfeng Zhan, Si-Min He, and Zhifei Zhang. pdeep: predicting ms/ms spectra of peptides with deep learning. Analytical chemistry, 89(23):12690–12697, 2017.

[51] Yuhao Zhou, Yiheng Wang, Xuming He, Ruoyao Xiao, Zhiwei Li, Qiantai Feng, Zijie Guo, Yuejin Yang, Hao Wu, Wenxuan Huang, et al. Scientists' first exam: Probing cognitive abilities of mllm via perception, understanding, and reasoning. arXiv preprint arXiv:2506.10521, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 abstract and introduction clearly state the main contributions: the proposal of CrossNovo, a hybrid AT-NAT framework for de novo peptide sequencing; its key innovations (shared encoder, dual decoders with cross-decoder attention, specialized training strategy with importance annealing and gradient blocking); and its superior performance over AT and NAT baselines on a 9-species benchmark. These claims are reflected in the methods section (Section 3) and experimental results (Section 4, Tables 1, 2, Figure 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 paper includes a "Limitations" section in Appendix A. It discusses that while the 9-species benchmark is extensive, it does not cover all phylogenetic diversity, and performance on highly distinct or rarely investigated species remains an avenue for future work.

   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 proposes a novel architecture and empirical results. While it uses established mathematical formulations for its components (e.g., attention, loss functions described in Section 3 and Appendix B), it does not present new theoretical results in the form of theorems or lemmas requiring formal proofs.

   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: Section 4.1 "Experiments Setup" details the datasets used (MassIVE-KB, 9-species-v1, 9-species-v2), implementation details (embedding sizes, layer numbers, attention heads, hidden dimensions, optimizer, learning rate, GPU type), evaluation metrics, and baselines. Section 3 provides a detailed description of the model architecture and training methods. Appendix sections further elaborate on CTC loss (B), PMC decoding (C), and beam size effects (D). This information should be sufficient to reproduce the main experimental results.

   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 abstract states, "Our code is available for reproduction: Link Here." The datasets used (MassIVE-KB, 9-species benchmarks, antibody datasets) are publicly available and cited (Section 4.1, Section 4.2, Appendix E). Instructions for reproducing results would presumably be part of the provided code repository.

   - 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: Section 4.1 details the datasets for training (MassIVE-KB) and validation/testing (9-species-v1, 9-species-v2). Hyperparameters (embedding dimensions, layers, heads, hidden dimensions, learning rate, optimizer AdamW, LR schedule) are specified. The choice of beam size is justified in Section 4.2 and Appendix D. How other hyperparameters were chosen (e.g., grid search) is not explicitly stated, but the values used are 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: [No]

   Justification: While the paper presents extensive empirical comparisons across multiple benchmarks, each primary experiment was conducted a single time to obtain the final results due to cost consideration.

   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: Section "Implementation Details" mentions the experiments compute resources.

   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 paper conform 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: This paper discuss positive and negative societal impacts in the Appendix.

    Guidelines:

    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 develops a model for peptide sequencing, which does not fall into the category of high-risk generative models or datasets requiring specific misuse safeguards as described in the guidelines.

    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: All datasets and baseline models are properly cited, with their respective licenses acknowledged in the text.

    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: New assets introduced in the paper are well documented in the paper.

    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 research is focused on developing a computational model for analyzing biological data (mass spectra) and does not involve crowdsourcing or direct research with human participants.

    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: As the research does not involve human subjects, IRB approval is not applicable.

    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: The paper does not involve the use of Large Language Models (LLMs) as a core method.

    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.

A Appendix

A.1 Limitations

The current study validated CROSSNOVO using an extensive 9-species benchmark, which provides substantial evidence for its robust performance and generalization capabilities across these organisms. However, a focused limitation of this work is that our evaluation, while covering multiple species, did not extend to an exhaustive representation of the vast phylogenetic diversity present across all kingdoms of life. The species included in the benchmark, though diverse, may not fully encapsulate all proteomic complexities or unique peptide sequence characteristics that could be present in organisms from exceptionally divergent evolutionary lineages or those adapted to extreme and understudied environments. Consequently, while CROSSNOVO demonstrates strong cross-species performance within the considerable scope of the current benchmarks, its specific performance nuances when applied to proteomes from such highly distinct or rarely investigated species remain an avenue for potential future investigation. This would serve to further confirm the breadth of its applicability across the widest possible biological landscape.

A.2 CTC Loss for Protein Sequence Prediction

Connectionist Temporal Classification (CTC) loss [10] is a pivotal objective function for sequence-to-sequence modeling, especially when the precise alignment between input (e.g., mass spectra $\mathcal{I}$) and output (e.g., peptide sequence $\mathcal{A}$) is unknown or variable. Our approach leverages CTC to train a model that predicts amino acid sequences from spectral data without requiring explicit alignment supervision. This section details the CTC formulation as employed in our work.

A.2.1 Problem Definition

Given an input spectrum $\mathcal{I}$ and a target amino acid sequence $\mathcal{A} = (a_1, a_2, \ldots, a_U)$, where U is the length of the target peptide, CTC aims to maximize the sum of probabilities of all possible alignment paths (or "blank-augmented" sequences) $\mathbf{z} = (z_1, z_2, \ldots, z_T)$ of length T (typically the length of the model's output feature sequence) that can be reduced to $\mathcal{A}$. The reduction operation $\mathcal{B}(\mathbf{z})$ involves merging consecutive identical non-blank tokens and removing all blank tokens (ϵ).

The conditional probability of the target sequence $\mathcal{A}$ given the input $\mathcal{I}$ is:

$$P(\mathcal{A}|\mathcal{I}) = \sum_{\mathbf{z}:\mathcal{B}(\mathbf{z})=\mathcal{A}} P(\mathbf{z}|\mathcal{I}). \tag{8}$$

Assuming conditional independence of outputs given the input at each time step t:

$$P(\mathbf{z}|\mathcal{I}) = \prod_{t=1}^{T} P(z_t|\mathcal{I})_t, \tag{9}$$

where $P(z_t|\mathcal{I})_t$ is the probability of observing token z_t at time step t, provided by the neural network. Direct summation over all paths is intractable. Thus, we employ a dynamic programming approach.

A.2.2 Dynamic Programming Formulation

Let $\mathcal{A}' = (\epsilon, a_1, \epsilon, a_2, \ldots, \epsilon, a_U, \epsilon)$ be the target sequence $\mathcal{A}$ interspersed with blanks at the beginning and between every token, resulting in a sequence of length $U' = 2U + 1$. We define a forward variable $\alpha_t(s)$ as the total probability of all paths that emit the prefix $\mathcal{A}'_{1:s}$ using the first t time steps of the model output.

The initialization for $t = 1$ is:

$$\alpha_1(1) = P(z_1 = \epsilon|\mathcal{I})_1 \tag{10}$$

$$\alpha_1(2) = P(z_1 = a_1|\mathcal{I})_1 \tag{11}$$

$$\alpha_1(s) = 0, \quad \forall s > 2. \tag{12}$$

The recursion for $t > 1$ and $s \geq 1$ is:

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1))\, P(z_t = \mathcal{A}'_s|\mathcal{I})_t, \tag{13}$$

if $\mathcal{A}'_s = \epsilon$ or $\mathcal{A}'_s = \mathcal{A}'_{s-2}$ (allowing repeats of characters if separated by a blank). If $\mathcal{A}'_s \neq \epsilon$ and $\mathcal{A}'_s \neq \mathcal{A}'_{s-2}$ (i.e., $\mathcal{A}'_s$ is a distinct character that must be taken, or the previous character in $\mathcal{A}'$ was different):

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2))\, P(z_t = \mathcal{A}'_s|\mathcal{I})_t. \tag{14}$$

Appropriate boundary conditions (e.g., $\alpha_t(0) = 0$ for $t > 0$, and $\alpha_{t-1}(s-2) = 0$ if $s < 2$) must be handled.

The total probability $P(\mathcal{A}|\mathcal{I})$ is the sum of probabilities of paths ending in either the last blank $\mathcal{A}'_{U'}$ or the last amino acid $\mathcal{A}'_{U'-1}$ at time T:

$$P(\mathcal{A}|\mathcal{I}) = \alpha_T(U') + \alpha_T(U'-1). \tag{15}$$

A.2.3 Loss Function

The CTC loss is the negative log-likelihood of this probability:

$$\mathcal{L}_{\text{CTC}} = -\log P(\mathcal{A}|\mathcal{I}). \tag{16}$$

This formulation enables end-to-end training of our peptide sequencing model by efficiently marginalizing over all possible alignments. Numerical stability is maintained by performing computations in log-space.

A.3 Novel Precise Mass Control (PMC) Decoding for Non-Autoregressive Transformers

We introduce Precise Mass Control (PMC), a novel knapsack-style dynamic programming algorithm specifically engineered for decoding peptide sequences from non-autoregressive transformer (NAT) models. A core challenge in applying NATs to *de novo* peptide sequencing is ensuring that generated sequences adhere to strict physical and experimental constraints. PMC addresses this by guaranteeing that the decoded peptide sequence $\mathcal{P}$ rigorously matches the experimentally observed precursor mass m_{pr} within a user-defined tolerance δ_{mass}. This direct integration of mass constraints into the decoding process is crucial for *de novo* sequencing, as it substantially prunes the vast search space of possible amino acid combinations, thereby significantly enhancing prediction accuracy and chemical validity. PMC offers a principled way to reconcile the parallel output generation of NATs with the precise, sequential constraints inherent in peptide mass spectrometry.

A.3.1 Problem Formulation

Let $P_t(y|\mathcal{I})$ denote the probability distribution over the vocabulary of amino acids $\mathcal{V}_{AA}$ (augmented with a blank token ϵ) for each position $t \in \{1, \ldots, T_{\text{seq}}\}$, as predicted by a NAT sequence model conditioned on input $\mathcal{I}$. Given a target precursor mass m_{pr} and a mass tolerance δ_{mass}, the objective of PMC is to find a peptide sequence $\mathcal{P} = (p_1, \ldots, p_L)$. This peptide $\mathcal{P}$ is derived by applying a CTC-like collapse function, $\mathcal{B}(\cdot)$, to an underlying path $\mathbf{y} = (y_1, \ldots, y_{T_{\text{seq}}})$, such that $\mathcal{P} = \mathcal{B}(\mathbf{y})$. The goal is to maximize the sum of log-probabilities of the path $\mathbf{y}$:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \sum_{t=1}^{T_{\text{seq}}} \log P_t(y_t|\mathcal{I})$$

subject to the mass constraint:

$$m_{\text{pr}} - \delta_{\text{mass}} \leq \sum_{j=1}^{L} u(p_j) \leq m_{\text{pr}} + \delta_{\text{mass}}, \tag{17}$$

where $u(p_j)$ represents the monoisotopic mass of the amino acid p_j in the peptide $\mathcal{P}$.

A.3.2 Mass Discretization for Dynamic Programming

To facilitate the dynamic programming approach, continuous mass values are discretized. We define a discretization function $f_{\text{disc}}(\cdot)$ that maps a continuous mass to a discrete mass bin index. Consequently, the monoisotopic mass of each amino acid $aa \in \mathcal{V}_{AA}$ is mapped to its discretized counterpart, $u'(aa) = f_{\text{disc}}(u(aa))$. The target precursor mass m_{pr} is also discretized to $m'_{\text{pr}} = f_{\text{disc}}(m_{\text{pr}})$, and the maximum allowable discretized mass for any peptide prefix is denoted M'_{max}. The tolerance δ_{mass} defines a target mass range $[m'_{\text{lower}}, m'_{\text{upper}}]$ where $m'_{\text{lower}} = f_{\text{disc}}(m_{\text{pr}} - \delta_{\text{mass}})$ and $m'_{\text{upper}} = f_{\text{disc}}(m_{\text{pr}} + \delta_{\text{mass}})$.

A.3.3 Dynamic Programming State

The core of PMC is a DP table, D. An entry $D[t][m]$[last_y_ne] stores the maximum accumulated log-probability of a path prefix $y_1, \ldots, y_t$ such that:

1. The CTC-collapsed peptide derived from $y_1, \ldots, y_t$ has a total discretized mass m.
2. The last non-ϵ token encountered in the path $y_1, \ldots, y_t$ was last_y_ne $\in \mathcal{V}_{AA} \cup \{\text{null}\}$. The null value is used for initialization or if all preceding tokens were ϵ.

Storing last_y_ne is essential for correctly applying CTC collapse rules, particularly for handling repeats and insertions. Each entry $D[t][m]$[last_y_ne] stores a tuple: $(\text{log\_probability}, \text{collapsed\_peptide\_sequence})$. While we describe the top-1 (Viterbi) version for clarity, this DP formulation can be extended to beam search by storing the top-B candidates in each cell.

Initialization At $t = 0$, before processing any tokens: $D[0][0][\text{null}] = (0.0, \text{empty\_sequence})$. All other entries $D[0][m]$[last_y_ne] are initialized to $(-\infty, \text{empty\_sequence})$. The mass $m = 0$ corresponds to an empty peptide.

Recursion The DP table is populated iteratively for each time step $t = 1, \ldots, T_{\text{seq}}$: For each previous discretized mass $m_{old} \in \{0, \ldots, M'_{\text{max}}\}$: For each previous last non-ϵ token prev_y_ne $\in \mathcal{V}_{AA} \cup \{\text{null}\}$: If $D[t-1][m_{old}]$[prev_y_ne].log_probability $> -\infty$ (i.e., state is reachable): Let $(\text{logP}_{old}, \text{peptide}_{old}) = D[t-1][m_{old}]$[prev_y_ne]. For each token $y_t \in \mathcal{V}_{AA} \cup \{\epsilon\}$ with probability $P_t(y_t|\mathcal{I})$: Let $\text{logP}_{cand} = \text{logP}_{old} + \log P_t(y_t|\mathcal{I})$.

1. **If $y_t = \epsilon$ (blank token):** The collapsed peptide and its mass remain unchanged. The last non-ϵ token also remains prev_y_ne.
   - $m_{new} = m_{old}$
   - $\text{peptide}_{new} = \text{peptide}_{old}$
   - next_y_ne = prev_y_ne

   If $\text{logP}_{cand} > D[t][m_{new}]$[next_y_ne].log_probability, update $D[t][m_{new}]$[next_y_ne] $=$ $(\text{logP}_{cand}, \text{peptide}_{new})$.
2. **If $y_t \in \mathcal{V}_{AA}$ and $y_t =$ prev_y_ne (repeat of last non-ϵ token):** This corresponds to a stutter in the y-sequence (e.g., $A \to AA$). The CTC-collapsed peptide and its mass do not change. The last non-ϵ token is updated to y_t.
   - $m_{new} = m_{old}$
   - $\text{peptide}_{new} = \text{peptide}_{old}$
   - next_y_ne $= y_t$

   If $\text{logP}_{cand} > D[t][m_{new}]$[next_y_ne].log_probability, update $D[t][m_{new}]$[next_y_ne] $=$ $(\text{logP}_{cand}, \text{peptide}_{new})$.
3. **If $y_t \in \mathcal{V}_{AA}$ and $y_t \neq$ prev_y_ne (new amino acid added to peptide):** The amino acid y_t is appended to the collapsed peptide. Its mass is added. The last non-ϵ token becomes y_t.
   - $m_{new} = m_{old} + u'(y_t)$
   - $\text{peptide}_{new} = \text{peptide}_{old} \circ y_t$ (where $\circ$ denotes concatenation)
   - next_y_ne $= y_t$

   If $m_{new} \leq M'_{\text{max}}$ and $\text{logP}_{cand} > D[t][m_{new}]$[next_y_ne].log_probability, update $D[t][m_{new}]$[next_y_ne] $= (\text{logP}_{cand}, \text{peptide}_{new})$.

A.3.4 Final Sequence Selection

After populating the DP table up to $t = T_{\text{seq}}$, the optimal peptide sequence $\mathcal{P}^*$ is determined. We iterate through all possible final non-ϵ tokens last_y_ne_final $\in \mathcal{V}_{AA} \cup \{\text{null}\}$ and all discretized masses m_{final} such that $m'_{\text{lower}} \leq m_{final} \leq m'_{\text{upper}}$. The peptide sequence $\mathcal{P}^*$ is the collapsed_peptide_sequence from the entry $D[T_{\text{seq}}][m_{final}]$[last_y_ne_final] that has the highest log_probability among these candidates.

This PMC algorithm uniquely embeds precursor mass constraints directly within a CTC-compatible decoding framework for NATs. By ensuring adherence to experimental observations, PMC significantly refines the output of NATs for challenging tasks like *de novo* peptide sequencing, leading to more accurate and physically plausible molecular identifications.

A.4 Influence of Various Beam Sizes

Based on the experimental results presented in Table 5, increasing the beam size generally enhances both Amino Acid Precision and Peptide Recall across various species. At both the amino acid and peptide levels, accuracy tends to improve with larger beam sizes, though it stabilizes later, exhibiting only minor increments or remaining unchanged.

For Amino Acid Precision, increasing the beam size from 1 to 3 significantly boosts average precision from 0.784 to 0.804, a rise of 0.020. Further increases up to a beam size of 11 yield smaller gains, with the highest precision of 0.811 observed at beam sizes of 5 and 11, but the rate of improvement diminishes. In terms of Peptide Recall, increasing the beam size from 1 to 3 raises average recall from 0.634 to 0.651, an increase of 0.017. Beyond a beam size of 3, improvements are marginal, with the highest recall of 0.654 achieved at beam sizes of 5, 7, and 9. Some species exhibit slight recall decreases at larger beam sizes, likely due to the Seq2Seq model's exposure bias [29, 44, 22]. The model, trained with Teacher Forcing, consistently receives the correct prior output during training but must generate its own during inference, leading to potential deviations from the optimal solution. Larger beam sizes can mitigate this issue, but excessively large sizes might cause overfitting and hinder generalization.

While larger beam sizes can enhance prediction performance, they also increase inference costs. To balance effectiveness and speed, we selected a beam size of 5 for our experiments. With this size, the model achieves high performance metrics, with an average precision of 0.811 and an average recall of 0.654, showing minimal differences compared to larger beam sizes. Additionally, compared to beam sizes of 9 or 11, a beam size of 5 offers faster inference speed and lower computational demands, maintaining prediction performance while optimizing efficiency. Therefore, considering both performance and computational costs, a beam size of 5 is considered optimal, achieving an effective balance between accuracy and efficiency.

Table 5: Comparison of Amino Acid Precision and Peptide Recall for 9-Species-v1 at Various Beam Sizes

Metrics	Species	1-Beam	3-Beam	5-Beam	7-Beam	9-Beam	11-Beam
Amino Acid Precision	Bacillus	0.829	0.850	0.856	0.854	0.854	0.855
	Clambacteria	0.713	0.728	0.740	0.734	0.734	0.734
	Honeybee	0.758	0.779	0.785	0.785	0.785	0.786
	Human	0.766	0.792	0.800	0.800	0.802	0.802
	M.mazei	0.801	0.819	0.826	0.824	0.825	0.824
	Mouse	0.794	0.813	0.816	0.816	0.816	0.817
	Ricebean	0.793	0.820	0.831	0.827	0.828	0.828
	Tomato	0.812	0.826	0.830	0.830	0.832	0.832
	Yeast	0.791	0.809	0.814	0.815	0.815	0.816
	Average	0.784	0.804	**0.811**	0.810	0.810	**0.811**
Peptide Recall	Bacillus	0.706	0.725	0.726	0.727	0.726	0.726
	Clambacteria	0.502	0.517	0.518	0.519	0.519	0.518
	Honeybee	0.591	0.607	0.610	0.610	0.610	0.610
	Human	0.632	0.657	0.661	0.663	0.664	0.663
	M.mazei	0.642	0.658	0.660	0.660	0.660	0.660
	Mouse	0.577	0.593	0.596	0.596	0.596	0.595
	Ricebean	0.686	0.712	0.716	0.717	0.717	0.717
	Tomato	0.682	0.694	0.695	0.694	0.694	0.694
	Yeast	0.684	0.696	0.698	0.698	0.698	0.698
	Average	0.634	0.651	**0.654**	**0.654**	**0.654**	0.653

A.5 Precision-Coverage Curves

To evaluate the efficacy of our model, we utilize Precision-Coverage curves, which offer insights into performance across various species. A full visual representation of CrossNovo's outstanding performance is depicted in the Precision-Coverage curve shown in Figure 1. The horizontal axis represents coverage, while the vertical axis represents peptide recall. The blue line indicates the performance of Casanovo V2, the orange line represents ContraNovo, and the green line shows the performance of our model. Across all subplots, the green lines are consistently positioned above the blue and orange lines, illustrating the superior performance of our model in peptide recall over varying coverage levels. This consistent outperformance suggests potential for more accurate peptide identification, which could enhance biological insights.

B A.7 Cross-Attention Alignment Analysis

In our extended analysis of the interaction between the Autoregressive Transformer (AT) decoder and the Non-Autoregressive Transformer (NAT) feature map, we observed a consistent and interpretable alignment pattern. Specifically, when the AT decoder generates a token at position t, its cross-attention weights are predominantly concentrated around the corresponding t-th token region in the NAT feature representation. This spatial-temporal correspondence suggests that the AT decoder has learned to selectively focus on the NAT's bidirectional contextual embedding at each generation step, rather than distributing attention uniformly across the entire sequence.

Such behavior provides strong empirical evidence that the NAT decoder functions as a powerful auxiliary encoder, supplying global contextual information that complements the AT's locally conditioned generation process. Through this mechanism, the AT decoder effectively integrates the NAT's bidirectional representation as a guiding signal, thereby enhancing both token-level consistency and long-range structural coherence in sequence generation.

B.1 Downstream Tasks

Table 6: Comparison of the performance of CrossNovo and baseline methods on WIgG1-Mouse. The bold font indicates the best performance.

Metrics	Methods		HC		LC	Average
		AspN	Chymotrypsin	Trypsin	AspN	
Amino Acid Precision	Casa.V2	0.714	0.591	0.723	0.668	0.674
	Contra.	0.750	0.612	0.650	0.649	0.665
	Ours	**0.769**	**0.640**	**0.747**	**0.724**	**0.720**
Peptide Recall	Casa.V2	0.557	0.483	0.636	0.456	0.533
	Contra.	0.649	0.545	0.671	0.519	0.596
	Ours	**0.662**	**0.577**	**0.699**	**0.581**	**0.630**

Table 7: Comparison of the performance of CrossNovo and baseline methods on IgG1-Human. Bold values indicate the best performance.

Metrics	Methods				HC			LC		Average
		AspN	Chymo.	GluC	LysC	Proteinase	Trypsin	AspN	LysC	
Amino Acid Precision	Casa.V2	0.520	0.472	0.605	0.757	0.354	0.759	0.666	0.778	0.642
	Contra.	0.580	0.565	0.642	0.790	0.348	0.787	0.702	0.793	0.676
	Ours	**0.613**	**0.617**	**0.694**	**0.814**	**0.367**	**0.803**	**0.719**	**0.807**	**0.702**
Peptide Recall	Casa.V2	0.265	0.274	0.399	0.569	0.206	0.595	0.325	0.625	0.446
	Contra.	0.396	0.372	0.437	0.653	0.274	0.675	0.499	0.646	0.529
	Ours	**0.415**	**0.421**	**0.512**	**0.701**	**0.275**	**0.699**	**0.544**	**0.676**	**0.560**

Data. The Human IgG1 antibody dataset (IgG1-Human) consists of mass spectrometry data collected using the LTQ Orbitrap instrument. Ionization was performed through higher-energy collisional dissociation (HCD), and the resulting peptide fragments were captured at a resolution of 17,500 FWHM. The dataset features peptides digested by a variety of proteolytic enzymes, including trypsin, chymotrypsin, asp-N, lys-C, glu-C, and proteinase K. We perform the evaluation on donwloaded data with no processing.

The Mouse IgG1 antibody dataset (WIgG1-Mouse) was similarly analyzed using an LTQ Orbitrap mass spectrometer with HCD ionization and the same resolution of 17,500 FWHM. In this dataset, the mouse peptides were digested with trypsin, asp-N, and chymotrypsin to generate a comprehensive proteomic profile. We also used downloaded data for evaluation of all tested models.

Results. As shown in Tables 7 and 6, CrossNovo consistently achieves superior performance across both amino acid-level precision and peptide-level recall when compared to the baseline methods. Specifically, for the Mouse dataset, CrossNovo demonstrates a notable improvement in peptide recall, achieving up to a 6% increase for the AspN enzyme on the light chain (LC) protein. Similarly, in the Human dataset, the peptide recall improvement is up to 4.5% for the AspN enzyme on the LC protein.

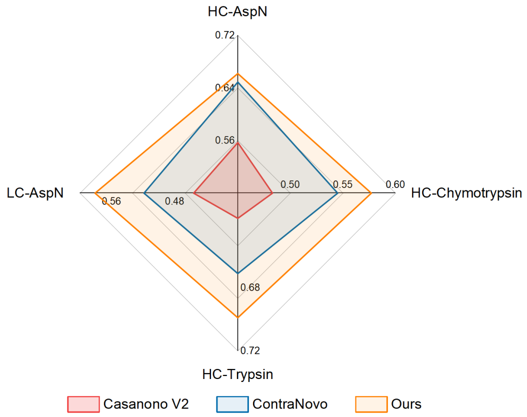

Figure 5: The comparison of performance of models in mouse antibody data.

The performance gain is particularly pronounced for light chain proteins across both species, with CrossNovo showing higher overall precision and recall. The average amino acid precision for the Mouse dataset reaches 0.720, while the peptide recall is boosted to 0.630. For the Human dataset, CrossNovo attains an average precision of 0.702, with a peptide recall of 0.560, further underscoring its effectiveness over baseline approaches.

These differences in performance are even more apparent in Figure 5, where CrossNovo's enhancements, particularly on the light chain proteins, can be clearly visualized. The consistency in performance improvements across both datasets highlights CrossNovo's ability to handle diverse proteolytic enzymes with high accuracy, especially in cases involving the light chains.

B.2 Post-Translational Modification (PTM) Fine-Tuning Evaluation

To further evaluate the generalization ability of CrossNovo and PrimeNovo, we extend our experiments to post-translational modification (PTM) prediction tasks. Each PTM model was fine-tuned independently on the curated **21-PTM dataset**, containing spectra annotated with 21 types of peptide modifications. For conciseness, we report results on the first five PTMs in alphabetical order: Acetylation, Biotinylation, Crotonylation, Butyrylation, and Dimethylation.

Analysis. As shown in Table, both models maintain high classification accuracy after PTM-specific fine-tuning. CrossNovo consistently achieves superior or comparable results across all metrics, particularly in amino acid and peptide recall for complex modifications like Butyrylation and Dimethylation. This demonstrates the advantage of cross-decoder attention and bidirectional contextual transfer in recognizing subtle mass shifts introduced by PTMs. Furthermore, fine-tuning on the

Table 8: Performance comparison on five representative PTMs after fine-tuning on the 21-PTM dataset. Bold values indicate the best performance for each column.

Metric	**Model**	**Acetylation**	**Biotinylation**	**Crotonylation**	**Butyrylation**	**Dimethylation**
Classification Accuracy	PrimeNovo	0.98	**0.99**	0.98	0.97	0.95
	CrossNovo	**0.98**	0.98	**0.98**	**0.98**	**0.97**
AA Recall	PrimeNovo	0.95	**0.83**	**0.90**	0.82	0.84
	CrossNovo	**0.96**	**0.83**	0.86	**0.85**	**0.85**
Peptide Recall	PrimeNovo	0.89	0.70	**0.79**	0.65	0.66
	CrossNovo	**0.90**	**0.72**	0.78	**0.69**	**0.68**

diverse 21-PTM dataset highlights the scalability of our architecture to biochemical modifications, extending its applicability beyond canonical peptide sequencing.

B.3 Broader Impact

Advancements in *de novo* peptide sequencing have far-reaching societal implications, particularly in healthcare, biotechnology, and life sciences. CrossNovo significantly enhances the accuracy and robustness of sequence prediction, directly benefiting applications such as neoantigen discovery for personalized cancer immunotherapy, rapid characterization of therapeutic antibodies, and profiling of microbial communities in environmental and clinical metaproteomics. By improving the capacity to identify novel or mutated peptides without reliance on existing databases, our method contributes to overcoming current bottlenecks in drug discovery, vaccine development, and disease diagnostics. Moreover, CrossNovo's hybrid architecture—balancing predictive stability with rich contextual understanding—enables more reliable deployment in translational pipelines where interpretability and robustness are critical. However, as with any powerful bioinformatic tool, careful consideration must be given to issues of data privacy, equitable access to computational resources, and responsible clinical translation to ensure these technologies are used ethically and inclusively.