INSearch: AI-native framework for large scale proteomic database search via contrastive joint embeddings and transformer-based scoring function

Tasneem Midhat Mustafa Attaallah (tasneem@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr Konstantinos Kalogeropoulos
Technical University of Denmark, Denmark
Co-supervised by Mr Jeroen Van Goey
InstaDeep, Cape Town Office

June 29, 2026
Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

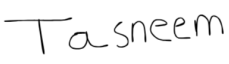
In mass spectrometry-based proteomics, identifying the peptide sequence that generated a specific spectrum is a complex challenge. The main method that is used for peptide-spectrum matching is searching against a reference database using traditional search engines; these search engines use a tailored score function to assign a score to each match based on certain criteria. However, as datasets grow in size, the search space expands combinatorially due to the inclusion of many organism proteomes or post-translational modifications, and the time complexity of these search engines is $O(S \times \rho P)$ (Kalogeropoulos et al., 2026), where S and P are the number of query spectra and peptides, respectively.

The aim of this project is to design a preliminary framework prototype called INSearch. The INSearch objective is to provide a sublinear candidate retrieval complexity $O(S(\log P + K))$ while efficiently maintaining identification accuracy in large-scale database searches. INSearch utilises a dual-encoder architecture to jointly project the experimental tandem mass spectra and theoretical peptide sequences into a shared latent space so that the spectra and their corresponding ground truth peptides align. The standard contrastive loss was used as the objective loss function with an additional component, variance regularisation (Bardes et al., 2022), to prevent feature collapse. Another loss function was used as an alternative to the standard contrastive loss, alignment and uniformity loss (Wang and Isola, 2020) that optimises the contrastive learning objective asymptotically. The similarity search is conducted through an approximate nearest neighbour search algorithm to retrieve the top-k peptide candidates for a certain spectrum query, providing a sublinear time complexity. Since the similarity search is approximate, the retrieved candidate list undergoes a neural re-ranking stage: the decoder of InstaNovo (Eloff et al., 2025) is used in teacher-forcing mode as a scoring function to assign residue-specific log-probabilities, which are aggregated using the geometric mean into a final confidence score for each match.

On the nine-species benchmark, the two-stage INSearch achieved strong recall on the held-out yeast test split: a Stage-1 Recall@$1/5/100$ of $60.0/78.9/91.9\%$, with the neural re-scoring stage lifting Recall@1 to 82.9%, indicating cross-species generalisation. On the more challenging *S. brodae* proteome the results were more moderate (a Stage-1 Recall@100 of 55.5%), indicating a further need for scaling and training on massive datasets.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work and that any work done by others or by myself previously has been acknowledged and referenced accordingly.



Tasneem Midhat Mustafa Attaallah, June 29, 2026

Contents

List of Abbreviations

Abbreviation	Meaning
AdamW	Adam optimiser with decoupled weight decay
AI	Artificial Intelligence
AIMS	African Institute for Mathematical Sciences
AMP	Automatic Mixed Precision
ANN	Approximate Nearest-Neighbour
CLIP	Contrastive Language-Image Pre-training
DNS	De Novo Sequencing
FAISS	Facebook AI Similarity Search
FDR	False Discovery Rate
FP16	16-bit (Half-Precision) Floating Point
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HCD	Higher-energy Collisional Dissociation
HNSW	Hierarchical Navigable Small World
IN	InstaNovo
InfoNCE	Information Noise Contrastive Estimation
k-NN	k-Nearest Neighbours
MLP	Multilayer Perceptron
MoCo	Momentum Contrast
MS	Mass Spectrometry
m/z	Mass-to-Charge ratio
PSM	Peptide-Spectrum Match
PTM	Post-Translational Modification
SimCLR	Simple Framework for Contrastive Learning of Visual Representations
TDC	Target–Decoy Competition
t-SNE	t-Distributed Stochastic Neighbour Embedding
VICReg	Variance–Invariance–Covariance Regularisation
vRAM	Video Random-Access Memory
XCorr	Cross-Correlation (score)

1. Introduction

1.1 Background

Mass spectrometry (MS) is the core technology for peptide identification in the field of proteomics, enabling the characterisation of proteins in complex samples and providing insight into the composition, structure, function and control of the proteome (Aebersold and Mann, 2016). In bottom-up proteomics, the central task is to infer peptide sequences from tandem mass spectra, typically by matching observed spectra against theoretical spectra generated from an in silico digestion of a protein database. There are two primary methods to solve the peptide assignment problem: database search and de novo sequencing. Traditional database search uses a tailored score function and subsequent re-scoring using spectrum features to evaluate the quality of a peptide-spectrum match (PSM), whereas state-of-the-art de novo sequencing (DNS) uses a deep learning approach to solve the peptide assignment problem.

1.2 Problem statement

Accurately producing and evaluating peptide-spectrum matches remains computationally demanding because spectra are noisy, incomplete, and highly variable, potentially leading to errors in the resulting database search algorithm scores (Venable and Yates, 2004). Additionally, traditional scoring functions use fragmentation matches between the experimental spectra and the theoretical ones; as database size increases, the number of random matches will increase, thus negatively impacting the reliability of these methods. This challenge intensifies in searches that must account for post-translational modifications, where the search space expands combinatorially, and conventional scoring methods become less effective and fail to scale efficiently. Therefore, the main bottleneck with traditional database search methods is scalability when querying multiple spectra against millions of peptides in a database and not having a scoring function that is well-calibrated or discriminative enough to retrieve more spectrum-peptide matches based on fine-grained discrimination.

1.3 Recent deep learning methods

Recent deep learning methods have substantially advanced proteomic spectrum analysis. Transformer-based models (Eloff et al., 2025) frame DNS as a sequence-to-sequence translation problem, mapping fragmentation patterns directly to amino acid sequences without requiring database priors. In parallel, retrieval models such as yHydra (Altenburg et al., 2021) learn joint representations of tandem mass spectra and peptides in a shared embedding space, enabling efficient similarity search through simple distance computations and GPU-accelerated nearest neighbour retrieval over large databases.

1.4 Aim and objectives

Specifically, this project builds upon the previous advancements and pursues the following objectives:

- To build a preliminary prototype that uses supervised contrastive learning and offers a principled way to learn spectrum-peptide joint embeddings in a latent space by mapping matched peptide-spectrum pairs together while separating mismatched pairs.
- To propose a solution for the scalability bottleneck through approximate nearest neighbour search

that enables fast and efficient similarity search with a small, tolerable trade-off between speed and accuracy.

- To re-score/re-rank the top retrieved candidates through the decoder of a transformer-based de novo sequencer operating in teacher-forcing mode to generate per-residue scores for more accurate and well-calibrated ranking of the candidates.

2. Literature review and theoretical background

2.1 Literature review

The traditional database search solves the peptide identification problem by searching for an observed tandem MS spectrum against a pre-defined set of peptides in a database using empirical score functions to score the top candidate peptides within a certain m/z range. Therefore, the query spectrum is compared against the theoretical spectrum of each candidate peptide that falls within this pre-specified m/z range. These score functions, conceptually, share the same framework of measuring the number of shared fragment ion peaks between the observed spectrum and the theoretical spectrum for each peptide, while differing in their implementation details. The most popular classical database search engines are: SEQUEST (Eng et al., 1994), Mascot (Perkins et al., 1999), MS-GF+(Kim and Pevzner, 2014), X!Tandem(Craig and Beavis, 2004), and MSFragger (Kong et al., 2017), however, their associated score functions have been demonstrated to be poorly calibrated (Keich and Noble, 2015) and can't capture the intricate fragmentation patterns underlying the peptides measured (Zolg et al., 2018) (Ananth et al., 2024) as well as missing low intensity fragment ions taking into account the most intense b and y ions. Additionally, database search is limited to peptide sequences present in the database, thus hindering the identification of protein isoforms and engineered or evolved proteins without prior sequence knowledge, and it is agnostic to transcription or translation errors (Eloff et al., 2025), this is crucial for infected cells, detection of pathogens, and metaproteomics applications where there are many organisms in the same sample. Another major limitation of database search is the high computational costs at scale; the computational complexity scales as $O(S \times \rho P)$, where S denotes the number of query spectra, P the number of database peptides, and ρ the fraction of peptides whose theoretical precursor masses fall within the pre-specified precursor mass tolerance window. Additionally, when accounting for post-translational modifications in an open search setting, the search space increases combinatorially, further complicating the search space complexity problem (Kalogeropoulos et al., 2026).

Three families of deep-learning approaches have emerged to address these limitations, each tackling a different aspect of the problem.

Encoder–decoder transformer models such as DeepNovo (Tran et al., 2017), Casanovo (Yilmaz et al., 2024), and InstaNovo (Eloff et al., 2025) bypass the database entirely and predict peptide sequences directly from spectra as a sequence-to-sequence translation task. These models implicitly learn the relationship between fragmentation patterns and peptide sequences from millions of annotated PSMs, and their runtime scales as $O(S)$ in the number of spectra, independent of database size. Their main weakness is that any predicted sequence absent from the reference database becomes hard to interpret biologically, and inference cost per spectrum is much higher than classical scoring (Kalogeropoulos et al., 2026).

A complementary line of work re-purposes pretrained sequencing models as scoring functions rather than as generators. Casanovo-DB (Ananth et al., 2024) feeds the ground-truth candidate peptide back into the Casanovo decoder under teacher forcing and uses the geometric mean of per-residue log-probabilities as a PSM score. At a 1% FDR threshold, this learned score detects between 31–40% more peptides than XCorr, 52–69% more than Hyperscore, and 88–102% more than Andromeda across yeast, *E. coli*, and human benchmarks. These methods improve the *accuracy* of database search but not its *scalability*: scoring still requires evaluating one neural forward pass per candidate peptide, which scales as $O(S \cdot \rho P \cdot C_{\text{score}})$ and is several orders of magnitude slower than dot-product comparisons at large P.

A third family attacks the scaling problem head-on by training two encoders to map spectra and peptides into a shared embedding space, after which candidates are retrieved by approximate nearest-neighbour (ANN) search rather than by exhaustive scoring. yHydra (Altenburg et al., 2021) was the first such system, training paired transformer encoders on ≈ 20 million PSMs from 67 proteomics repositories with a CLIP-style contrastive loss (Radford et al., 2021) and using GPU-accelerated k-NN search to perform open searches. Their reported runtime is roughly $1.6\times$ faster than MSFragger at comparable identification rates, and the joint embedding makes open-PTM search natural because modifications appear as small displacements in embedding space rather than combinatorial database expansion. SpecEmbedding (Xiong et al., 2025) recently showed that supervised contrastive learning yields more discriminative spectral embeddings even in the single-modality metabolomics setting. (Kalogeropoulos et al., 2026) formalise the asymptotic cost as $O(S(\log P + K))$, sublinear in database size P, and identify bi-encoder + ANN as the only family that scales gracefully to databases of 10^8 peptides and beyond.

INSearch is a two-stage retrieval system that combines approximate nearest neighbour search with re-scoring using InstaNovo. The dual-encoder is composed of the spectrum encoder built around the *pretrained* InstaNovo transformer backbone rather than trained from scratch and a peptide encoder that is trained from scratch. Second, the architecture is paired with a second-stage *learned-score* re-ranker (the InstaNovo decoder used as a scoring function) so that the fast ANN retrieval supplies a small candidate shortlist that a neural scorer can re-rank. The result is a retrieval system that is optimised for speed and accuracy.

2.2 Theoretical background

This section serves as an introduction to the mathematical and theoretical framework underlying the search pipeline components.

2.2.1 Contrastive Representation Learning

Learning representations of data is about extracting relevant and useful information that can be used for a variety of downstream tasks (Bengio et al., 2013). Contrastive learning is a framework with origins in metric learning, which gained significant prominence as a self-supervised learning technique following the seminal works of SimCLR (Chen et al., 2020) and MoCo (He et al., 2019). The core objective is to learn an embedding space in which data points belonging to the same class are pulled closer together, while dissimilar data points are simultaneously pushed apart. More precisely, the goal is to learn the geometry of an embedding space that encodes both the coarse semantic structure of groupings and the fine-grained instance-based discrimination between individual samples.

Mathematically, given a set of input data points $\{x_i\}$, the objective is to learn a function parameterised by weights $f_\theta(\cdot) : \mathcal{X} \to \mathbb{R}^d$, referred to as a feature extractor or encoder, that maps each data point to an embedding vector of dimensionality d, such that similar points have proximal embeddings and dissimilar points have distant embeddings in the resulting embedding space. Contrastive learning can be applied in both self-supervised and supervised settings. In the supervised multimodal setting, it is used to learn joint embeddings across different data modalities that differ in format or structure, enabling cross-modal retrieval and alignment.

The InfoNCE Loss

Several loss functions have been proposed to achieve the contrastive training objective. One of the most widely adopted is the Information Noise Contrastive Estimation loss (InfoNCE) (Oord et al., 2018), defined as:

$$\mathcal{L} = -\log \frac{\exp\left(\text{sim}(z_i, z_j^+)/\tau\right)}{\sum_{k=1}^{N} \exp\left(\text{sim}(z_i, z_k)/\tau\right)} \tag{2.2.1}$$

where:

- z_i is the anchor embedding, the representation of the query data point.
- z_j^+ is the positive pair embedding, corresponding to a data point that belongs to the same class as the anchor or represents the ground truth match.
- z_k are the in-batch negative embeddings, all other samples in the training batch that are not the positive pair for the anchor, serving as implicit negatives under the assumption that randomly sampled pairs are unlikely to be true matches.
- $\text{sim}(\cdot)$ is a similarity function measuring the alignment between two embedding vectors, typically cosine similarity.
- τ is a temperature hyperparameter that controls the concentration of the distribution, with lower values producing sharper contrast between positives and negatives, and higher values producing a smoother, more uniform distribution.

The Role of Negatives

The denominator of the InfoNCE loss (Equation 2.2.1) aggregates similarity scores over all negatives in the batch, meaning that both the number and the quality of negatives directly determine the difficulty and informativeness of the learning signal. With a batch size of N, only $N - 1$ negatives are available per anchor at each training step. When negatives are sampled randomly, they are often trivially easy to discriminate from the anchor, providing a weak gradient signal and slowing convergence.

Hard negative mining addresses this by explicitly selecting negatives that are semantically close to the anchor but belong to different classes, thereby increasing the difficulty of the contrastive task and improving the quality of the learned representations. This is particularly consequential when the embedding space contains compositionally similar instances (such as peptide sequences differing by a single amino acid substitution or post-translational modification) where the boundary between true positives and hard negatives is fine-grained.

Application to Peptide-Spectrum Matching

In the context of peptide-spectrum matching (PSM), contrastive learning is applied in a supervised multimodal setting to learn a joint embedding space over two structurally distinct modalities: tandem mass spectra and peptide sequences. A spectrum encoder and a sequence encoder are trained jointly such that a spectrum and its correct peptide sequence are mapped to proximal regions of the shared embedding space, while spectra and incorrect peptide sequences are mapped to distant regions. This enables efficient similarity-based retrieval at inference time, given a query spectrum, the correct peptide can be identified by searching for its nearest neighbour in the pre-encoded peptide embedding space, without requiring explicit pairwise comparison against every candidate in the database.

Alignment and Uniformity

(Wang and Isola, 2020) provide a geometric interpretation of what contrastive learning optimises by decomposing the properties of a well-structured embedding space into two measurable objectives: alignment and uniformity.

Alignment measures how close positive pairs are mapped in the embedding space, quantifying the encoder's ability to produce invariant representations for semantically equivalent inputs:

$$\mathcal{L}_{\text{align}} = \underset{(x,y)\sim p_{\text{pos}}}{\mathbb{E}} \left[\|f(x) - f(y)\|_2^2 \right] \tag{2.2.2}$$

A lower alignment loss indicates that positive pairs are consistently mapped to proximal regions of the embedding space.

Uniformity measures how uniformly the embeddings are distributed across the unit hypersphere, quantifying the encoder's ability to preserve maximal information without representation collapse:

$$\mathcal{L}_{\text{uniform}} = \log \underset{\substack{\text{i.i.d} \\ x,y \sim p_{\text{data}}}}{\mathbb{E}} \left[e^{-t\|f(x)-f(y)\|_2^2} \right] \tag{2.2.3}$$

A lower uniformity loss indicates that embeddings are spread evenly across the space, maximising discriminative capacity between distinct instances.

These two objectives exist in tension: optimising alignment alone would collapse all embeddings to a single point, while optimising uniformity alone would distribute embeddings without regard to semantic similarity. Effective contrastive learning requires balancing both. Notably, (Wang and Isola, 2020) demonstrate that the InfoNCE loss implicitly optimises both objectives simultaneously, providing a theoretical justification for its empirical success.

Dimensional collapse

Even when full *representational* collapse is avoided, contrastive encoders frequently exhibit a milder pathology: the learned embeddings span only a small subspace of the available d dimensional space, with most coordinates carrying negligible variance across the dataset. (Jing et al., 2021) term this phenomenon *'dimensional collapse'* and trace it to two driving forces: strong augmentation pushing covariances toward zero along certain directions and implicit regularisation in the optimisation dynamics. The practical consequence is that the encoder's discriminative capacity is limited to the number of *effectively* used dimensions, often far fewer than the nominal embedding size. The effective rank (Roy and Vetterli, 2007), defined as the exponential of the entropy of the normalised singular-value spectrum of the embedding matrix, gives a continuous and differentiable diagnostic for this collapse and is the metric used in this work (Section 3.5).

VICReg (Bardes et al., 2022) addresses dimensional collapse directly by augmenting the contrastive objective with a per-dimension variance term that penalises any embedding coordinate whose standard deviation across a batch falls below a fixed threshold. The variance term provides an explicit gradient signal to inactive dimensions that alignment alone cannot supply, and is one of the regularisers we apply in Section 3.5.

2.2.2 Transformer architectures and the InstaNovo backbone

Both encoders in this work are transformer-based, and the spectrum encoder additionally re-uses a frozen pretrained backbone. We briefly review the relevant ingredients.

The transformer encoder (Vaswani et al., 2017) replaces the recurrent and convolutional inductive biases of earlier sequence models with a single primitive, scaled dot-product self-attention. Given a

sequence of input embeddings $X \in \mathbb{R}^{n \times d}$, learnable linear projections produce queries, keys and values $(Q, K, V) = (XW_Q, XW_K, XW_V)$, and the attended output is

$$\text{Attention}(Q, K, V) \;=\; \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{2.2.4}$$

where d_k is the key dimensionality. Multi-head attention applies this operation in parallel across h learned subspaces and concatenates the results. The attention mechanism is permutation-equivariant by construction, which makes it a natural fit for mass spectra: the set of fragment peaks has no canonical ordering, and the model is expected to learn whichever relationships between peaks are diagnostic of peptide identity. Positional information is injected separately, typically through sinusoidal positional embeddings on the input.

For mass spectra, the input to the transformer is the set of $(m/z, \text{intensity})$ tuples for each peak. InstaNovo and yHydra both use a sinusoidal positional embedding on the m/z axis concatenated with the (preprocessed) intensity, projected through a small MLP into the model dimension. The crucial property is that the sinusoidal encoding produces vectors that vary smoothly with m/z, so that fragments at near-identical m/z produce near-identical embeddings, a desirable inductive bias given the finite mass resolution of the instrument.

InstaNovo (Eloff et al., 2025) is an encoder–decoder transformer trained on ≈ 35 million PSMs across nine species for the de novo sequencing task. Its encoder produces contextualised peak representations of dimensionality $d = 768$, and its decoder generates peptide sequences autoregressively right-to-left under a cross-attention mechanism conditioned on the encoder output. In this work, the encoder is re-used as a *frozen* feature extractor for the spectrum side of the dual-encoder retriever, and the decoder is re-used as a *teacher-forced scorer* for the second-stage re-ranker. Freezing the backbone is empirically necessary: training the spectrum encoder from scratch on the available data produced severe dimensional collapse, while the frozen InstaNovo representations served as a robust starting point that a small projection head could re-shape for the contrastive objective.

2.2.3 Approximate nearest-neighbour retrieval

Once the dual encoder has been trained, inference reduces to a nearest-neighbour search: given a query spectrum embedding $z_s \in \mathbb{R}^d$ and a database of peptide embeddings $\{z_p^{(j)}\}_{j=1}^{P}$, return the top-k peptides maximising $\langle z_s, z_p^{(j)} \rangle$. Exact search is $O(P)$ per query and becomes prohibitive at the database sizes (10^7–10^9 peptides) considered in proteomics.

Hierarchical Navigable Small World (HNSW) (Malkov and Yashunin, 2018) is a graph-based approximate-nearest-neighbour data structure that organises the database into a hierarchy of proximity graphs of decreasing density. Each layer is a navigable small-world graph in which long-range links shortcut across the embedding space; queries descend the hierarchy layer by layer, refining the candidate set greedily until a target neighbourhood size is reached. FAISS (Johnson et al., 2021) provides a widely used implementation that we adopt in this work.

Since both encoders apply L_2normalisation to their outputs (Section 2.2.2), cosine similarity reduces to a plain inner product: $\cos(z_s, z_p) = \langle z_s, z_p \rangle$. This is convenient because most ANN libraries optimise the inner-product or L_2distance metrics, and the normalisation step is paid once at index-construction time rather than at every query.

2.2.4 Neural Re-scoring

A re-ranker (or re-scorer) is a model that re-evaluates and re-orders the candidates returned by an initial retrieval stage, trading computational efficiency for higher ranking precision.

A hypothesis that was formulated by (Ananth et al., 2024) states that DNS models implicitly learn a score function that captures the relationship between spectra and peptides and can be used instead of hand-designed functions. A score function is a mathematical algorithm that is used to evaluate the quality and hence acts as a proxy for confidence score of a peptide-spectrum match. Deep learning models are trained on massive tandem mass spectrometry data, enabling the models to learn better mapping between complex fragmentation patterns and their associated peptide sequences; with a slight re-engineering of the models, they can be repurposed as a learned score function. Precisely, the decoder of the Casanovo deep learning model (Yilmaz et al., 2024) was used in a teacher-forcing mode instead of generating the whole sequence autoregressively; the whole sequence can be fed to the decoder at once to gather the per-amino acid log probabilities and yield a final sequence-level score. The Casanovo-DB score function showed impressive results against traditional scoring functions over three benchmarks at a 1% FDR; they observed an average 88% improvement over the Andromeda score (E.coli: 88%, yeast: 75%, human: 102%), 57% improvement over the Hyperscore (E.coli: 69%, yeast: 50%, human: 52%), and a 35% improvement over XCorr (E.coli: 31%, yeast: 35%, human: 40%) (Ananth et al., 2024) in the number of identified PSMs.

Therefore, their findings suggest that de novo sequencing models can be repurposed as a scoring function, but the open question is whether they should be fine-tuned for database search or not; more information is in the future work section.

2.2.5 False discovery rate estimation by target–decoy competition

A PSM ranking by itself is not a usable identification report: the question is which calls can be trusted at a given error tolerance. The standard answer in proteomics is target–decoy competition (TDC), introduced by (Elias and Gygi, 2007).

For each target peptide in the search database, a corresponding *decoy* peptide is constructed by some null-preserving transformation that preserves overall sequence statistics (amino-acid composition, length distribution, terminal-residue identity in tryptic searches). Common strategies include sequence reversal, shuffling, and the *reverse-inner* strategy used in this work, which reverses only the internal residues and preserves both terminal residues. The combined target + decoy database is searched jointly, and the assumption is that an incorrect target match is statistically equivalent to a decoy match, so the rate of decoy top-1 hits at any score threshold serves as an unbiased estimator of the rate of incorrect target hits.

At score threshold τ, let $T(\tau)$ and $D(\tau)$ be the number of top-1 target and decoy matches with score $\geq \tau$. Under the equal-database-size assumption,

$$\widehat{\mathrm{FDR}}(\tau) \;=\; \min\!\left(1,\; \frac{D(\tau)}{T(\tau)}\right). \tag{2.2.5}$$

Sorting all PSMs by descending score and taking the running monotone minimum of $\widehat{\mathrm{FDR}}$ from the bottom of the list upward produces a *q-value* for each PSM: the smallest FDR at which that PSM would be accepted. The conventional reporting threshold in proteomics is 1% q-value.

The validity of the TDC estimator depends crucially on the score function being well calibrated across spectra: a target hit and a decoy hit at the same score must be equally likely to be incorrect. Hand-

designed scores such as XCorr and Hyperscore are known to be poorly calibrated in this sense (Keich and Noble, 2015), while learned scores trained on large PSM corpora have been shown to yield substantially better calibration (Ananth et al., 2024). This is one of the central motivations for replacing cosine similarity with the InstaNovo geometric-mean score in the second stage of our pipeline (Section 3.8).

3. Methodology

This section describes the methodologies used for developing the database search framework pipeline for peptide identification from experimental spectra. Methodological changes are made to dual-encoder training iteratively on the basis of the analysis of empirical signals from the training phase. The documentation process of these choices is necessary to provide information about the analysis of the results.

3.1 Pipeline overview

Before describing each component in detail, it is useful to see the whole pipeline at once. INSearch is a two-stage system: a fast dual-encoder augmented with ANN search retrieves a short candidate list for every query spectrum, and a neural re-scorer reorders that list conditioned on the spectrum query. The remainder of this chapter walks through each block in turn. The full pipeline is shown in Figure 3.1.

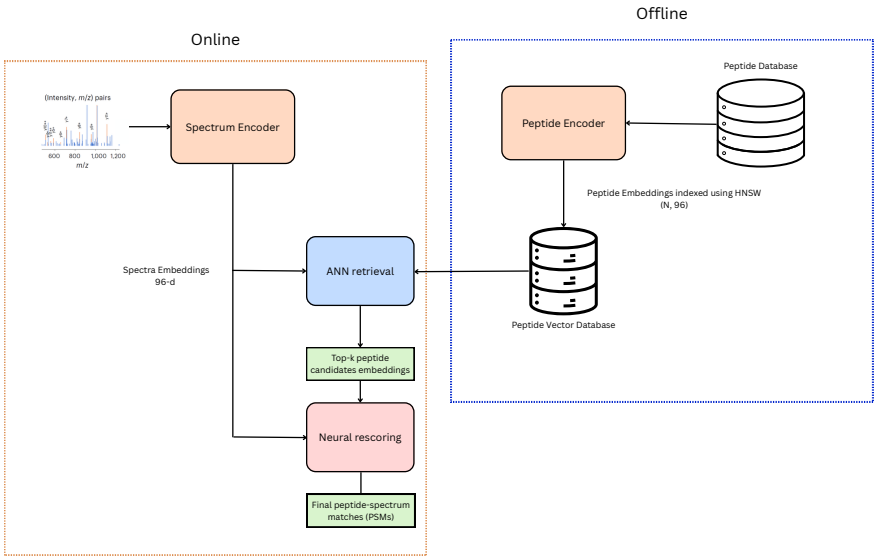


Figure 3.1: Overview of the two-stage INSearch pipeline. In the training phase, a spectrum encoder (a frozen InstaNovo backbone followed by a projection head) and a peptide encoder are jointly optimised with a contrastive objective so that a matching spectrum and peptide are mapped to nearby points in a shared embedding space. At deployment, the unique peptide database is embedded once and indexed with an HNSW graph. For each query spectrum, Stage 1 retrieves the top-K candidates by cosine similarity in $O(\log P)$ time, and Stage 2 re-scores these candidates with the InstaNovo decoder under teacher forcing, using the geometric mean of the per-residue probabilities.

1. **Training.** A spectrum encoder (frozen InstaNovo backbone + projection head) and a peptide encoder (transformer trained from scratch) are jointly optimised on the nine-species dataset with a contrastive objective. Both encoders output L_2-normalised embeddings in a shared d-dimensional space, so that cosine similarity reduces to a plain inner product. The training details are given in

Section 3.5; the loss-function iteration is documented in Section 3.4.

2. **Indexing.** At deployment time, the peptide database is deduplicated by modified-sequence string (Section 3.9) and the unique peptide embeddings are inserted into an HNSW approximate-nearest-neighbour index. The index is built once and reused across all query batches; details follow in Section 3.6.

3. **Stage-1 retrieval.** For each query spectrum, the spectrum encoder produces an embedding, and the HNSW index returns the top-K peptide candidates ranked by cosine similarity. This stage is cheap, its cost scales as $O(\log P)$ per query in the database size P, and is responsible for the pipeline's scalability.

4. **Stage-2 neural rescoring.** The top-K Stage-1 candidates for each spectrum are passed back through the pretrained InstaNovo decoder under teacher forcing, and the geometric mean of per-residue probabilities is used as a re-ranking score (Section 3.8). Reusing the spectrum encoder output across all K candidates amortises the encoder cost; only the decoder is run K times per spectrum.

3.2 Dataset and preprocessing

The publicly available nine-species benchmark dataset was used (Eloff et al., 2025). The dataset comprises tandem mass spectra acquired by higher-energy collisional dissociation (HCD) across nine species. Eight species were used for the training procedure, while *Saccharomyces cerevisiae* yeast was held out as a separate species used for validation and testing. The training set contains 499,402 peptide-spectrum matches (PSMs), and the held-out validation and test sets contain 28,572 and 111,312 PSMs respectively. The dataset contained other information such as precursor mass and charge that were used as additional signals during training; concatenated with the IN encoder output before being projected into the embedding space.

Each spectrum consists of two arrays of equal length: the mass-to-charge ratios (m/z) of the fragment peaks and their corresponding intensities. To match the maximum sequence length used during InstaNovo's pre-training, the number of peaks per spectrum was capped at 200; for spectra with more than 200 peaks, only the 200 most intense were retained, and shorter spectra were zero-padded. The intensity array was then preprocessed in two steps. First, each intensity value was square-root scaled, which compresses the dynamic range of fragment-ion intensities (typically spanning several orders of magnitude) and prevents a small number of dominant peaks from drowning out weaker but diagnostically informative fragments. Second, the square-rooted vector was divided by its L_2 norm, yielding a unit-norm intensity vector:

$$\tilde{I}_\ell = \frac{\sqrt{I_\ell}}{\sqrt{\sum_{k=1}^{N} I_k}}, \qquad \sum_{\ell=1}^{N} \tilde{I}_\ell^2 = 1. \tag{3.2.1}$$

This second step removes per-spectrum scale variability so that the encoder is presented with the *pattern* of fragmentation rather than its absolute magnitude. The resulting normalised intensities are then passed alongside the m/z values into the encoder's sinusoidal peak embedding, where each peak's m/z is encoded with a multi-scale sinusoidal positional embedding, concatenated with its scaled intensity, and projected through an MLP.

3.2.1 Handling post-translational modifications (PTMs)

The nine-species benchmark dataset contains spectra from peptides bearing some prevalent post-translational modifications. Each PTM was mapped to a specific token through the PTM-aware mapping set in the `ResidueSet` class provided by `InstaNovo` repository.

3.3 Dual Encoder Architecture

The dual encoder architecture was inspired by a CLIP-style (Radford et al., 2021) architecture, where two encoders should jointly map the spectra and the peptides into a shared d-dimensional latent embedding space. The two encoders produce ℓ_2 normalized embeddings, so that their normalized dot product (cosine similarity) reflects the similarity between these two vectors. The complete architecture is shown in Figure 3.2.

3.3.1 Spectrum Encoder

The spectrum encoder has undergone a major architectural change. Initially, the spectrum encoder was trained from scratch on the training dataset. The diagnostics showed problems regarding the representational collapse of the embeddings learned. Consequently, the pretrained InstaNovo (Eloff et al., 2025) encoder was utilised as the backbone to generate contextualised peak representations of dimensionality $d_{\text{InstaNovo}} = 768$. The encoder was held *frozen* during training, and only a projection head was trained to map the pooled peak representations into the d-dimensional latent embedding space. The projection head is composed of a three-layer multilayer perceptron with GELU activations, pre-layer normalization, and dropout:

$$g_\theta : \mathbb{R}^{768} \to \mathbb{R}^{1536} \to \mathbb{R}^{1536} \to \mathbb{R}^{d} \tag{3.3.1}$$

where d represents the dimensions that are used to encode the embeddings in the latent space and θ represents the parameters (weights) of the projection head. The final trainable projection head parameters were 3,692,640.

3.3.2 Peptide Encoder

The peptide encoder is a transformer trained from scratch using the set of canonical amino acids, rare amino acids and the post-translational modifications as the vocabulary for tokenization. Each residue, or amino acid is mapped to a unique index then used as a key for a learned amino acid embedding look-up table to retrieve the amino acid embedding. Positional information was added through a sinusoidal positional encoding. The encoder has four transformer encoder layers with $d_{\text{model}} = 128$, four attention heads and a feed-forward neural network with dimensionality $d_{\text{ff}} = 512$. A learned classification (CLS) token was prepended to the sequence, and after the transformer layers and a final layer normalization the contextualized representation at the CLS position was sliced out to serve as the final sequence representation. The sequence representation was projected onto the d- dimensional latent embedding space using a two-layer multilayer perceptron with GELU activations, pre-layer normalization, and dropout:

$$g_\theta : \mathbb{R}^{128} \to \mathbb{R}^{256} \to \mathbb{R}^{d} \tag{3.3.2}$$

where d represents the dimensions that are used to encode the embeddings in the latent space and θ represents the parameters (weights) of the projection head. The peptide encoder total parameters count is 1,119,072 parameters.

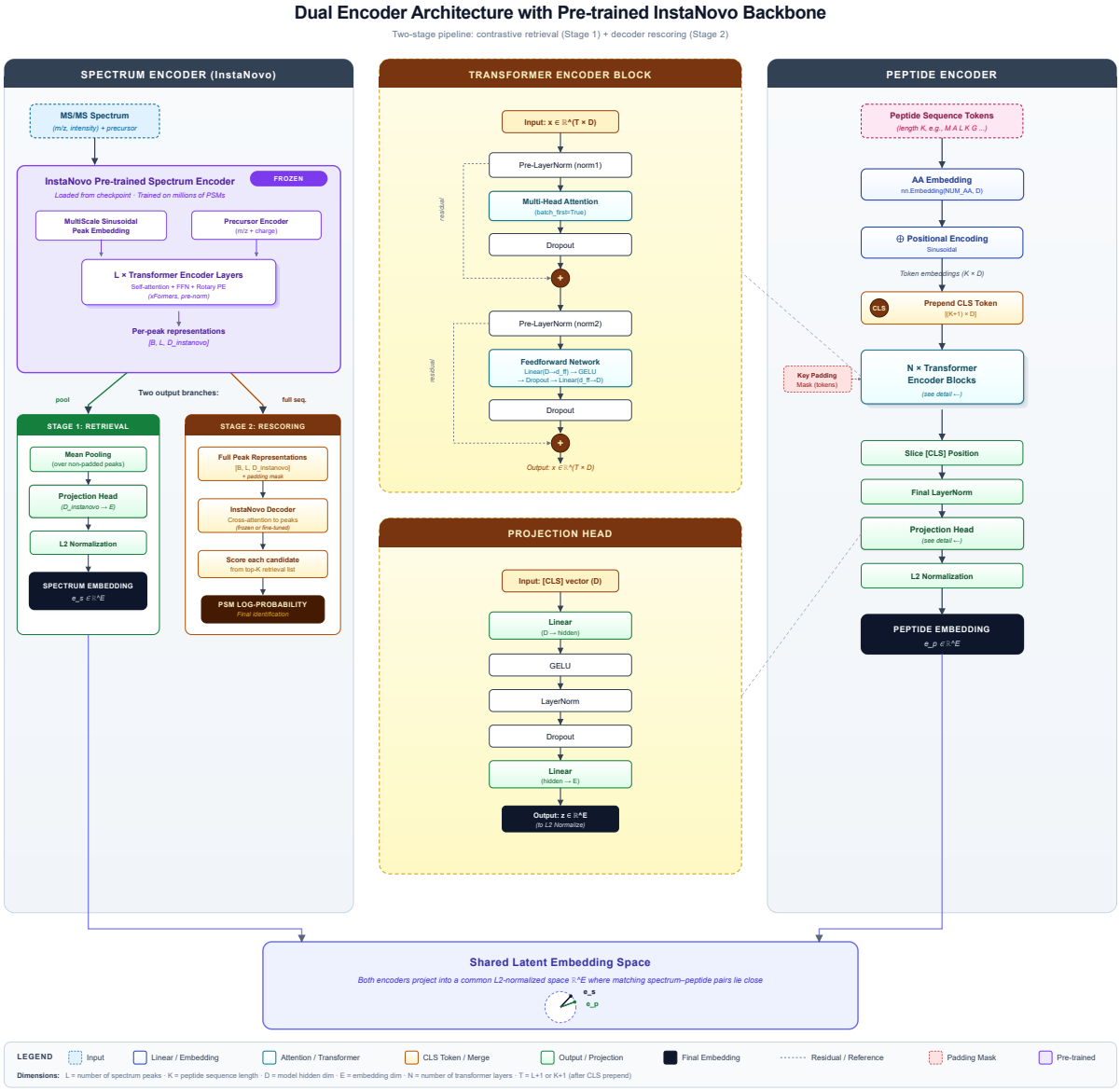


Figure 3.2: The INSearch dual-encoder architecture. *Left:* the spectrum encoder is built around the frozen, pretrained InstaNovo encoder, whose per-peak representations feed two branches: a mean-pooled, projected, and L_2-normalised *spectrum embedding* used for Stage-1 contrastive retrieval, and the full per-peak representations passed to the InstaNovo decoder for Stage-2 teacher-forced rescoring. *Right:* the peptide encoder, a transformer trained from scratch, maps a tokenised peptide sequence to an L_2-normalised *peptide embedding*. The two embeddings are trained jointly under the contrastive objective (bottom). The centre panels detail the shared transformer encoder block and the projection head.

3.4 Loss functions

The loss function went through an iterative adjustment process based on the evaluation and diagnostics results. Each change was documented and justified mathematically.

3.4.1 Symmetric Information Noise Contrastive Estimation (InfoNCE)

The infoNCE is the baseline loss function that is widely used in contrastive learning frameworks. In the CLIP paper (Radford et al., 2021), they used the symmetric infoNCE to calculate, first, the loss for each encoder separately and then calculate the average to yield the final loss used by the backpropagation algorithm to adjust the learnable parameters of each encoder. Because the dataset is labeled, which means each spectrum has the ground truth peptide sequence, it will facilitate the loss calculation in each direction where the spectrum and the ground truth peptide will be anchored against each other and the rest of the batch is considered negative samples.

$$\mathcal{L} = \tfrac{1}{2}\left(\mathcal{L}_{\mathsf{s}\to\,\mathsf{p}} + \mathcal{L}_{\mathsf{p}\,\to\,\mathsf{s}}\right) \tag{3.4.1}$$

$$\mathcal{L}_{\mathsf{s}\,\to\,\mathsf{p}} = -\tfrac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(z_i^s\cdot z_i^p/\tau)}{\sum_{j=1}^{N}\exp(z_i^s\cdot z_j^p/\tau)} \tag{3.4.2}$$

$$\mathcal{L}_{\mathsf{p}\,\to\,\mathsf{s}} = -\tfrac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(z_i^p\cdot z_i^s/\tau)}{\sum_{j=1}^{N}\exp(z_i^p\cdot z_j^s/\tau)} \tag{3.4.3}$$

Temperature annealing schedule

The temperature parameter τ controls the sharpness of the softmax distribution in the InfoNCE loss. Three schemes were evaluated: a learnable τ initialised at 0.07 following (Radford et al., 2021), a fixed $\tau = 0.5$, and a linear annealing schedule that decays τ from 0.5 to 0.07 over training (3.4.4). The annealing schedule was adopted for all subsequent experiments.

$$\tau(t) = \tau_{\mathsf{start}} + (\tau_{\mathsf{end}} - \tau_{\mathsf{start}})\cdot\frac{t}{T}, \quad \tau_{\mathsf{start}} = 0.5,\ \tau_{\mathsf{end}} = 0.1, \tag{3.4.4}$$

where T is the total number of epochs, and t is the current epoch.

Reversed-Peptide Decoy Strategy as an explicit hard negative

Inspired by the target–decoy strategy that underpins false discovery rate estimation in proteomics database search (Elias and Gygi, 2007), we introduced a reversed-peptide decoy as an explicit competitor in the contrastive loss. For each peptide p_i in the batch, a decoy $p_i^{(d)}$ is constructed by reversing the inner residues while preserving the C-terminal lysine or arginine (consistent with tryptic specificity):

$$p_i^{(d)} = \mathrm{reverse}(p_{i,1:L-1})\,||\,p_{i,L}. \tag{3.4.5}$$

The decoy is encoded through the peptide encoder and its similarity to the corresponding spectrum is appended as an additional column to the in-batch similarity matrix, augmenting it from $B\times B$ to $B\times(B+1)$.

The decoy carries identical amino acid composition, mass and terminal residues to the target, but a different fragmentation pattern. By including it as a competitor in the softmax denominator, the spectrum encoder is required to learn fragmentation-order sensitivity rather than relying on coarse compositional features alone.

Variance regularization

The effective rank (Roy and Vetterli, 2007), a measure of how many dimensions carry meaningful variance in the embedding vectors, remained low after the temperature annealing schedule: $20/96$. This is a

widely encountered phenomenon known as *dimensional collapse* (Jing et al., 2021), in which the embedding vectors span a lower-dimensional subspace instead of utilizing the full embedding dimensionality, limiting the model's capacity to learn discriminative features along the collapsed dimensions.

An explicit variance-preservation term inspired by VICReg (Bardes et al., 2022) was added to the contrastive loss Eq (3.4.1). It penalizes any embedding dimension whose standard deviation across a batch falls below a threshold $\gamma = 1/\sqrt{d}$:

$$\mathcal{L}_{\text{var}}(\mathbf{Z}) = \frac{1}{d}\sum_{k=1}^{d} \max(0,\ \gamma - \sigma_k(\mathbf{Z})), \tag{3.4.6}$$

where $\sigma_k(\mathbf{Z})$ is the empirical standard deviation of dimension k across the batch. This provides explicit gradient signal to inactive dimensions that the contrastive loss alone cannot activate. The variance loss was applied to both encoders with weight $\lambda_{\text{var}} = 25$, following the original VICReg value.

3.4.2 Alignment and Uniformity Loss

The final and most consequential change replaced the InfoNCE loss entirely with a direct decomposition of contrastive learning into its two component objectives, as derived by Wang and Isola (Wang and Isola, 2020):

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(s,p)\sim p_{\text{pos}}} \left[\|f_s(s) - f_p(p)\|_2^{\alpha}\right], \tag{3.4.7}$$

$$\mathcal{L}_{\text{unif}}^{(s)} = \log \mathbb{E}_{s,s' \overset{\text{iid}}{\sim} p_{\text{spectra}}} \left[e^{-t\|f_s(s)-f_s(s')\|_2^2}\right], \tag{3.4.8}$$

$$\mathcal{L}_{\text{unif}}^{(p)} = \log \mathbb{E}_{p,p' \overset{\text{iid}}{\sim} p_{\text{peptides}}} \left[e^{-t\|f_p(p)-f_p(p')\|_2^2}\right]. \tag{3.4.9}$$

The alignment term directly minimises the Euclidean distance between matched spectrum–peptide pairs on the unit hypersphere. The uniformity term, computed separately for each encoder, employs a Gaussian potential to drive the marginal distributions of embeddings toward uniformity on the unit hypersphere.

To incorporate domain knowledge from proteomics, an decoy margin loss was introduced; it is inspired by the classical target-decoy strategy (Elias and Gygi, 2007). For each spectrum s_i, a decoy peptide $\tilde{p}_i$ is constructed by reversing the target peptide sequence while preserving the C-terminal tryptic residue (K or R). The decoy loss penalises cases where the spectrum embedding is closer to its decoy than to its true match:

$$\mathcal{L}_{\text{decoy}} = \mathbb{E}_i \left[\max\left(0,\ \|f_s(s_i) - f_p(p_i)\|_2 - \|f_s(s_i) - f_p(\tilde{p}_i)\|_2 + \gamma\right)\right], \tag{3.4.10}$$

where $\gamma = 0.2$ is the margin hyperparameter.

The decoy embeddings are computed under `torch.no_grad()` to prevent contradictory gradients from flowing back through the shared peptide encoder parameters. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda\left(\mathcal{L}_{\text{unif}}^{(s)} + \mathcal{L}_{\text{unif}}^{(p)}\right)/2 + \lambda_{\text{decoy}}\mathcal{L}_{\text{decoy}}, \tag{3.4.11}$$

with $\alpha = 2$, $t = 2$, $\lambda = 1$, and $\lambda_{\text{decoy}} = 0.3$.

This formulation is theoretically motivated by the asymptotic decomposition of the InfoNCE loss into alignment and uniformity in the limit of infinite negative samples (Wang and Isola, 2020). With finite negatives (here $B = 128$), the contrastive loss is a biased approximation of the asymptotic objective. Direct optimisation of alignment and uniformity uses all $B(B-1)/2$ pairwise distances in the batch rather than $B - 1$ negatives per anchor, providing a denser estimate of the uniformity signal.

3.5 Training Procedure

All models were trained on a single NVIDIA A100 GPU using mixed-precision (FP16) arithmetic via the PyTorch automatic mixed precision (AMP) framework. The AdamW optimiser (Loshchilov and Hutter, 2019) was used with an initial learning rate of 3×10^{-4}, weight decay 10^{-2}, and a warmup–cosine learning rate schedule (10-epoch linear warmup followed by cosine decay to 10^{-6}). Training proceeded for 30 epochs with a batch size of 128 spectrum–peptide pairs. Gradient norms were clipped to a maximum of 1.0.

Each epoch logged training loss, in-batch accuracy, validation loss, validation accuracy, current temperature (when applicable), the variance loss components, and the mean positive–decoy similarity gap measured on the final batch of each epoch. After every training run, a full embedding-space audit was conducted on the training and held-out validation and test splits, comprising the alignment loss, the per-feature standard deviation, the within-encoder mean off-diagonal cosine similarity, the cross-encoder positive/negative similarity gap, the uniformity loss, and the effective rank computed via the entropy of the normalised singular value spectrum.

3.6 Indexing the peptide database with HNSW

At inference time, the trained dual encoder is used as a retrieval system: every test spectrum is encoded once, and the peptide database is searched for its nearest neighbours in the joint embedding space. Exact nearest-neighbour search is $O(P)$ per query in the database size P, which becomes a bottleneck at the scales considered in proteomics open search (tens of millions of peptides). To preserve the asymptotic $O(\log P)$ scaling that makes the bi-encoder family attractive in the first place (Kalogeropoulos et al., 2026), the peptide embeddings are stored in a Hierarchical Navigable Small World (HNSW) approximate-nearest-neighbour graph (Malkov and Yashunin, 2018), implemented in FAISS (Johnson et al., 2021).

HNSW organises the database as a hierarchy of proximity graphs of decreasing density. Each layer is a navigable small-world graph in which long-range links shortcut across the embedding space; queries descend the hierarchy layer by layer, refining the candidate set greedily until a target neighbourhood size is reached. The graph is parameterised by M, the maximum number of neighbours retained per node at each layer, and by two beam-search widths: $\mathrm{ef}_{\mathrm{construction}}$ used during graph build and $\mathrm{ef}_{\mathrm{search}}$ used at query time. Both control the speed/recall trade-off: larger values give higher recall at the cost of additional memory and latency.

Because both encoder heads apply L_2-normalisation to their outputs (see Equation (3.3.1)) and the peptide-encoder description), cosine similarity reduces to a plain inner product: $\cos(z_s, z_p) = \langle z_s, z_p \rangle$ for unit vectors. The index is therefore built with `faiss.IndexHNSWFlat` under the `METRIC_INNER_PRODUCT` metric, which lets the ANN library optimise the dot-product kernel directly without re-normalising at query time.

The training corpus contains many spectra per unique peptide. Inserting one embedding per row would waste top-k slots (two retrievals pointing to different rows that share the same sequence are semantically identical hits) and would distort ground-truth matching by row index. Prior to index construction, the peptide list is deduplicated by modified-sequence string: only the first occurrence of each unique peptide is retained, and a `row_to_unique` map records which spectra was paired with which unique-peptide id. Ground-truth matching downstream (Recall@k, FDR) is then performed by sequence-string equality, not by index row.

The values used throughout this work are

$$M = 32, \qquad \mathrm{ef}_{\mathsf{construction}} = 200, \qquad \mathrm{ef}_{\mathsf{search}} = 128, \qquad K = 100, \tag{3.6.1}$$

where K is the number of nearest neighbours requested per query. With $M = 32$. The query-time parameter $\mathrm{ef}_{\mathsf{search}}$ is runtime-tunable without rebuilding the index, which is useful when sweeping the recall/latency trade-off post hoc.

A built index is persisted to three files: the FAISS index itself, the integer id-map that translates internal FAISS row indices back to unique-peptide ids, and a JSON record of the configuration. The same index can then be reloaded across experiments without retraining the encoders or re-encoding the database.

3.7 Evaluation metrics

3.7.1 In-training evaluation metrics

For evaluating the dual-encoder, the average *in-batch accuracy* was computed on both the training and validation sets throughout training. The *in-batch accuracy* is calculated in both directions over a single batch. For both loss functions, the similarity matrix is first computed as the dot product of the spectra embeddings and the peptide embeddings, $Z_s \cdot Z_p^\top$, for a given batch. The diagonal entries of the similarity matrix represent the ground-truth pairings. For each row and each column, the index corresponding to the highest similarity score is extracted and compared against the diagonal index at that respective row or column. The number of correct predictions is then divided by the batch size.

$$\hat{j}_i = \operatorname*{argmax}_j \; S_{i,j} \tag{3.7.1}$$

$$\hat{i}_j = \operatorname*{argmax}_i \; S_{i,j} \tag{3.7.2}$$

$$\mathsf{Acc}_{\mathsf{spec}} = \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}\left(\operatorname*{argmax}_j \; S_{i,j} = i \right) \tag{3.7.3}$$

$$\mathsf{Acc}_{\mathsf{pep}} = \frac{1}{B} \sum_{j=1}^{B} \mathbb{I}\left(\operatorname*{argmax}_i \; S_{i,j} = j \right) \tag{3.7.4}$$

$$\mathsf{Acc}_{\mathsf{batch}} = \frac{\mathsf{Acc}_{\mathsf{spec}} + \mathsf{Acc}_{\mathsf{pep}}}{2} \tag{3.7.5}$$

3.7.2 Retrieval evaluation metrics

For evaluating the downstream retrieval task, Recall@k was used as the evaluation metric. For spectrum-level Recall@k, the following formula is used:

$$\mathsf{Recall@}k \;=\; \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big(r_i \in \mathcal{C}_k^{(i)} \Big), \tag{3.7.6}$$

where N is the total number of query spectra, r_i is the ground-truth peptide for the i-th spectrum, and $\mathcal{C}_k^{(i)}$ is the set of top-k retrieved candidate peptides for the i-th spectrum.

For the peptide-coverage Recall@k, the fraction of unique peptides in the database that are discovered across all query spectra is measured:

$$\text{Peptide Coverage Recall@}k \;=\; \frac{\left|\left(\bigcup_{i=1}^{N} \mathcal{C}_k^{(i)}\right) \cap \mathcal{P}\right|}{|\mathcal{P}|}. \tag{3.7.7}$$

where $\mathcal{P}$ is the set of all unique peptides in the database, $\mathcal{C}_k^{(i)}$ is the set of top-k retrieved candidates for the i-th spectrum, and $|\cdot|$ denotes set cardinality.

3.8 Neural rescoring

The dual-encoder retrieval provides a fast first-stage shortlist but is trained only with a contrastive objective and is therefore insensitive to fine-grained fragmentation evidence such as exact peak intensities and the presence or absence of individual b/y ions. Following Casanovo-DB (Ananth et al., 2024), the pretrained InstaNovo de novo sequencing transformer (Eloff et al., 2025) was re-purposed as a PSM scoring function in a second stage that re-ranks the Stage-1 shortlist.

3.8.1 Teacher-forced log-probability under InstaNovo

For a spectrum s and a candidate peptide sequence $p = (a_1, a_2, \ldots, a_L)$, the InstaNovo decoder is run under *teacher forcing*: rather than allowing the model to generate residues autoregressively, the ground-truth previous residues are fed at every position, and the decoder produces the distribution over the next residue in one forward pass. The per-residue log-probabilities

$$\ell_i = \log p_\phi(a_i \mid a_{<i},\ s), \qquad i = 1, \ldots, L, \tag{3.8.1}$$

are read out by gathering the appropriate column of the log-softmax of the decoder logits, where ϕ denotes the frozen parameters of the pretrained InstaNovo model. Because InstaNovo decodes right-to-left, candidate peptides are reversed and an end-of-sequence token is appended before being passed through the decoder.

3.8.2 Geometric-mean scoring

The per-residue log-probabilities are aggregated into a single PSM score by taking their mean,

$$\bar{\ell}(s, p) \;=\; \frac{1}{L} \sum_{i=1}^{L} \log p_\phi(a_i \mid a_{<i},\ s), \tag{3.8.2}$$

which is equivalent to the logarithm of the *geometric mean* of the per-residue probabilities. The final neural score is recovered by exponentiation,

$$\text{score}(s, p) \;=\; \exp(\bar{\ell}(s, p)) \;=\; \left(\prod_{i=1}^{L} p_\phi(a_i \mid a_{<i}, s)\right)^{1/L} \;\in\; [0, 1], \tag{3.8.3}$$

and is monotonically equivalent to the mean log-probability for ranking purposes. The geometric mean was preferred over the joint sequence log-probability $\sum_i \ell_i$ because the joint probability scales with peptide length and would bias the score against longer candidates (Ananth et al., 2024); the per-residue mean is length-normalised and penalises individual uncertain positions more strongly.

3.8.3 Two-stage retrieval and rescoring

The full pipeline operates as follows. For each query spectrum, the dual encoder retrieves the top-K candidates from the deduplicated peptide database under cosine similarity. Each of these K candidates is then scored against the spectrum using Eq. (3.8.3) with the InstaNovo decoder under teacher forcing, reusing the spectrum encoder output across candidates to amortise the encoder cost. The candidates are sorted in descending order of neural score, and this re-ranked head replaces the original Stage-1 head; the Stage-1 candidates beyond rank K are appended in their original cosine order so that Recall@k for $k > K$ cannot be degraded by the rescoring step.

3.9 False discovery rate estimation

Recall@k measures whether the ground-truth peptide is retrieved among the top-k candidates, but a deployable database-search tool must additionally answer the question of *which* top-1 calls can be trusted; thus, a measure of the model's reliability. Following standard practice in proteomics database search, false discovery rate (FDR) was estimated by target–decoy competition (TDC) (Elias and Gygi, 2007), in which a population of plausible-but-incorrect peptides (decoys) is generated through searched alongside the real target database, and the relative frequency of top-ranking decoy matches is used as an estimator of the proportion of incorrect target matches at any score threshold.

As in Section 3.6, the peptide database used for FDR estimation is the deduplicated set of unique modified sequences, and ground-truth matching downstream is by sequence-string equality rather than by index row.

3.9.1 Reverse-inner decoy construction

For each unique target peptide, a single token-level decoy was generated using the *reverse-inner* strategy. Given a tokenised target $p = (p_1, p_2, \ldots, p_{L-1}, p_L)$, the decoy is constructed as

$$p^{(d)} = (p_1,\ \mathrm{reverse}(p_{2:L-1}),\ p_L), \tag{3.9.1}$$

i.e. both terminal residues are kept fixed while the internal residues are reversed. Fixing the C-terminal residue preserves consistency with tryptic specificity (lysine or arginine), and fixing the N-terminal residue preserves the precursor-charge-bearing N-terminus. To guarantee a one-to-one target–decoy pairing and prevent score-distribution contamination, decoys that (i) coincide with their parent target after reversal (palindromic peptides), (ii) collide with another sequence already present in the target database, or (iii) duplicate a decoy already generated were excluded along with their parent target. Spectra whose target peptide failed to produce a valid decoy were removed from the FDR analysis set.

3.9.2 Neural target–decoy FDR

The neural score in Eq. (3.8.3) is used as the PSM score for FDR estimation in the full pipeline. For each spectrum, the stage-1 top-K candidates against the combined target+decoy database are rescored with InstaNovo; the highest neural-score candidate is taken as the top-1, and its target/decoy label is used in Eq. (2.2.5). Per-spectrum q-values are computed using the neural scores. The accepted PSM set at a chosen global FDR threshold (typically 1%) is then the set of spectra whose q-value lies below the threshold and whose top-1 match is a target. The final global FDR is estimated as: $\frac{N_{\text{decoy}}}{N_{\text{targets}}}$

4. Results and Discussion

This chapter presents and evaluates the INSearch pipeline across four questions: (1) does the pretrained InstaNovo backbone remove the failure modes observed when training from scratch? (2) which loss function yields stronger retrieval? (3) how well does the full two-stage pipeline perform, and where does it fail? (4) does the framework generalise to an unseen proteome at realistic database scale?

4.1 Backbone Ablation: Training from Scratch versus Frozen InstaNovo

To establish a baseline and characterise the failure modes of contrastive training on this task, a fully from-scratch dual encoder was trained first. Both encoders were lightweight transformers ($d_{\text{model}} = 128$, 4 heads, 4 layers, $d_{\text{ff}} = 512$), the embedding space was of size 96, and the symmetric InfoNCE loss function was used, as described in Subsection 3.4.1.

On an in-distribution random split the model was healthy (Top-$1/5/10$ of $32.0/60.8/70.3\%$ at 50K, val. acc. 91.6%), but on the out-of-distribution split (yeast held out) it plateaued at $8.3/28.1/38.8\%$ Top-$1/5/10$ over the 111K pool with a $\sim$31-point train/val gap. A systematic sweep over model capacity, warm-up length, precursor injection, and covariance regularisation did not move the ceiling, indicating a need for a larger dataset.

The central diagnostic concerned the learnable temperature τ of the InfoNCE softmax. With τ learnable, the optimiser drove it to its lower clamp, sharpening the softmax so that in-batch accuracy climbed above 90% while the positive-pair cosine similarity *fell* from 0.78 to 0.50; high accuracy was achieved through a sharpening shortcut rather than genuine alignment. Fixing $\tau = 0.5$ removed the shortcut and yielded the best alignment (pos. sim. $= 0.894$) but induced severe dimensional collapse (effective rank $14/96$). A linear annealing schedule, described in Eq. 3.4.4, decreases the temperature as a function of the number of epochs, which partially resolved the issue. The annealing schedule stabilised the training and activated more dimensions of the embedding space, achieving an effective rank of $20/96$, still not optimal, as the embedding vectors span a low-rank subspace instead of utilising the whole space.

These failure modes motivated replacing the from-scratch spectrum encoder with the frozen, pretrained InstaNovo backbone. Freezing the backbone removed dimensional collapse: the frozen representations served as a robust starting point, and only a small projection head needed to be trained. All subsequent experiments use this architecture.

4.2 Loss Function Comparison

With the frozen InstaNovo backbone fixed, two loss functions were compared: the symmetric InfoNCE loss with temperature annealing and a variance regularisation term (Section 4.2.1), and the alignment-and-uniformity decomposition augmented with a decoy margin term (Section 4.2.2). Both models use identical architecture and training procedure; only the objective differs.

4.2.1 InfoNCE with Temperature Annealing and Variance Regularisation

The hyperparameter settings for this configuration are given in Table 4.1.

Table 4.1: Hyperparameters for the contrastive loss configuration.

Hyperparameter	Value
D_MODEL	128
N_{HEADS}	4
D_{FFN}	768
N_{layers}	4
$EMBED\_DIM$	96
$DROP\_OUT$	0.1
learning rate	0.0003
label smoothing	0.01
initial temperature	0.5
final temperature	0.1
warmup epochs	10
training epochs	30
seed	42

The training exhibited clear stability and smooth convergence over the total number of epochs, as illustrated by the training and validation loss and accuracy curves in Figure 4.1.

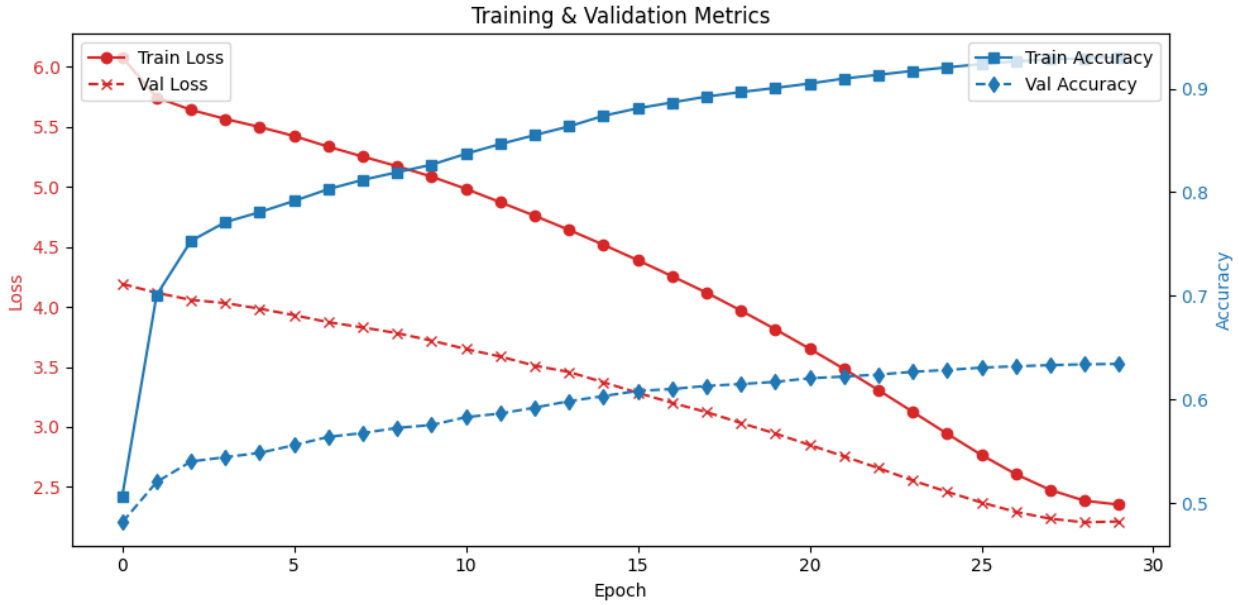


Figure 4.1: Training and validation metrics over the 30 epochs for the InfoNCE configuration.

The t-SNE visualisation in Figure 4.2 confirms that the model learned a semantically structured latent space, organised by precursor mass for both modalities.

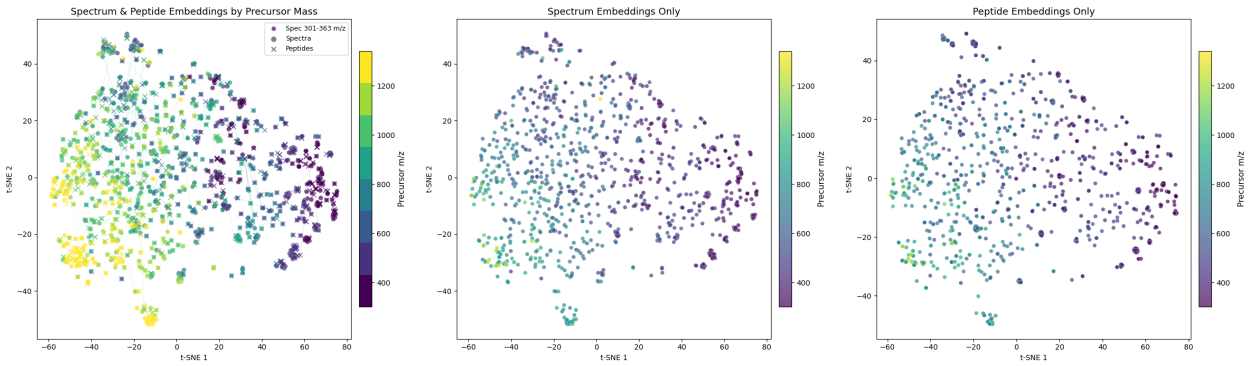


Figure 4.2: t-SNE visualisations of the spectrum and peptide embeddings produced by the InfoNCE model, coloured by precursor m/z value, for a randomly selected subset from the test dataset. *Left:* Joint visualisation of spectrum (circles) and peptide (crosses) embeddings, with lines connecting each spectrum to its corresponding ground truth peptide sequence. *Centre:* Spectrum embeddings only. *Right:* Peptide embeddings only.

4.2.2 Alignment and Uniformity Loss

The hyperparameters for this configuration are listed in Table 4.2.

Table 4.2: Hyperparameters for the alignment and uniformity loss configuration.

Hyperparameter	**Value**
D_MODEL	128
N_{HEADS}	4
D_{FFN}	768
N_{layers}	4
$EMBED\_DIM$	96
$DROP\_OUT$	0.1
learning rate	0.0003
warmup epochs	10
seed	42

Training was stable and converged smoothly, as shown in Figure 4.3. A gap between the training and validation curves is evident; because the validation set is the held-out yeast species, this reflects a cross-species generalisation problem rather than overfitting.

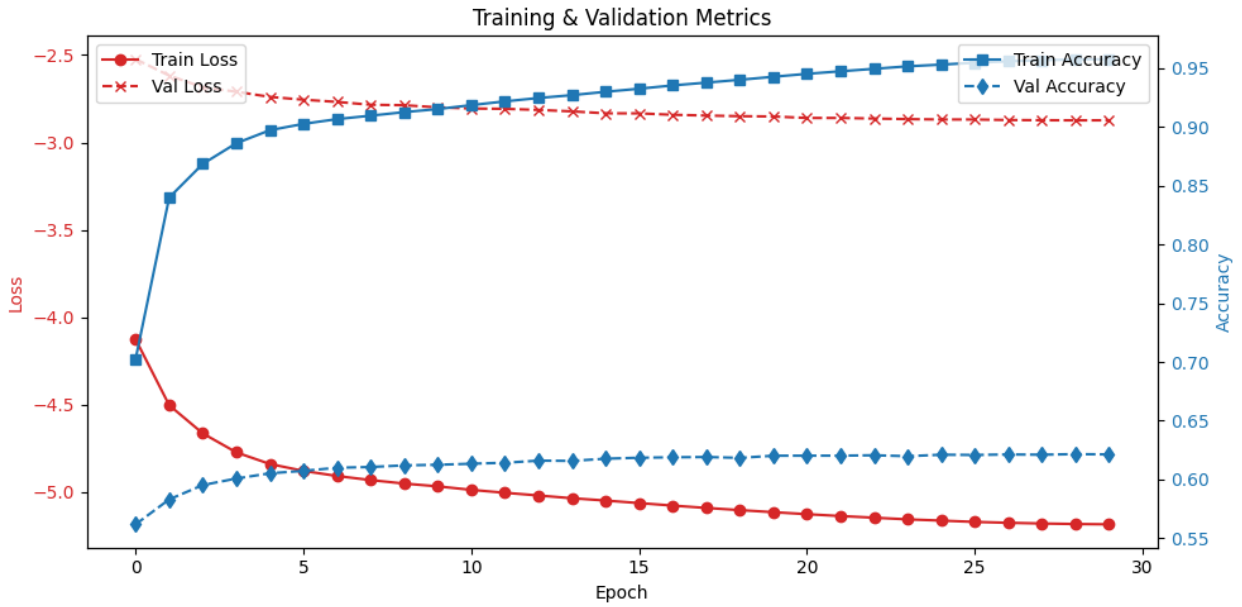


Figure 4.3: Training and validation loss and accuracy over 30 epochs for the alignment and uniformity configuration.

Figure 4.4 presents the corresponding t-SNE visualisations, which show a similar mass-organised structure to the InfoNCE model.

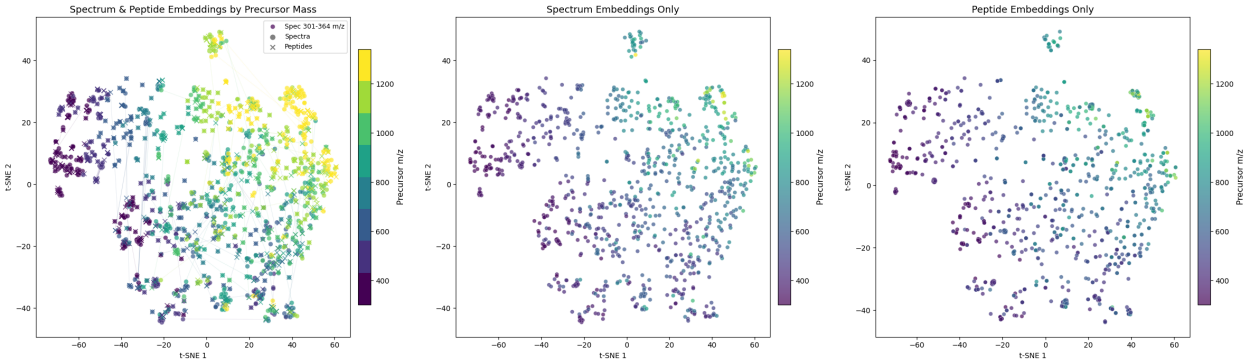


Figure 4.4: t-SNE visualisations of the spectrum and peptide embeddings produced by the alignment and uniformity model, coloured by precursor m/z value, for a randomly selected subset from the test dataset. *Left:* Joint visualisation of spectrum (circles) and peptide (crosses) embeddings, with lines connecting each spectrum to its corresponding ground truth peptide sequence. *Centre:* Spectrum embeddings only. *Right:* Peptide embeddings only.

4.2.3 Retrieval Performance Comparison

Table 4.3 reports Stage-1 Recall@k for both configurations on the held-out yeast test split ($n = 111{,}312$ spectra). The alignment-and-uniformity model outperforms the InfoNCE baseline across every retrieval depth. A note is that INSearch retrieves the top-100 candidates and then calculates the recall@k, and the re-scoring process is performed on the same top-100 candidates list. This introduces a hard ceiling on the recall@k values after the re-scoring— bounded by recall@100.

Table 4.3: Stage-1 HNSW retrieval performance comparison between the contrastive baseline (InfoNCE loss + variance term) and the alignment and uniformity model on the yeast test split ($n = 111{,}312$ spectra, seed 42). PSM-level Recall@k is computed over all query spectra. Peptide-level CandRecall@k reports the fraction of unique ground truth peptides ($N = 30{,}175$) recovered within the top-k candidates. Δ denotes the absolute improvement in percentage points.

Metric	**Contrastive loss**	**Alignment and Uniformity loss**	Δ
PSM-level Recall@k			
Recall@1	35.82%	**59.99%**	+24.17 pts
Recall@5	61.81%	**78.87%**	+17.06 pts
Recall@10	69.93%	**82.80%**	+12.87 pts
Recall@50	83.67%	**89.58%**	+5.91 pts
Recall@100	87.93%	**91.86%**	+3.93 pts
Not retrieved @100	13,438	**9,056**	−4,382 spectra
Median rank (found)	2	**1**	−1
Peptide-level CandRecall@k (unique peptides)			
CandRecall@1	47.40%	**66.57%**	+19.17 pts
CandRecall@5	68.34%	**80.81%**	+12.47 pts
CandRecall@10	74.62%	**84.03%**	+9.41 pts
CandRecall@50	85.47%	**89.71%**	+4.24 pts
CandRecall@100	89.07%	**91.72%**	+2.65 pts

The diminishing Δ as k increases indicates that the contrastive-only model was already placing correct peptides within the broad neighbourhood, but the alignment and uniformity objectives substantially improve local instance-based discrimination, producing tighter positive-pair alignment and more uniform separation of negatives. The alignment-and-uniformity model is therefore selected as the best-performing model for all subsequent stages.

Note that the two configurations differ in both the primary loss function and the decoy margin term; a granular ablation isolating the individual contribution of each component is left for future work.

Robustness Across Random Seeds. To assess sensitivity to stochastic variation in weight initialisation and data sampling order, the alignment-and-uniformity model was retrained with two additional random seeds. Table 4.4 reports Recall@k across all three seeds; standard deviations of 0.001–0.002 confirm that the results are stable and reproducible.

Table 4.4: Stage-1 HNSW Recall@k on the yeast test split across three random seeds. Results are reported as mean $\pm$ standard deviation.

k	Seed 42	Seed 0	Seed 123	Mean	Std
1	0.5999	0.6032	0.6021	0.6017	0.0017
5	0.7887	0.7921	0.7912	0.7907	0.0017
10	0.8280	0.8318	0.8299	0.8299	0.0019
50	0.8958	0.8982	0.8979	0.8973	0.0013
100	0.9186	0.9205	0.9199	0.9197	0.0010

4.3 Two-Stage Pipeline Performance

This section evaluates the complete INSearch pipeline on the held-out yeast test split, first assessing Stage-2 neural re-scoring and then characterising where the pipeline fails.

4.3.1 Neural Re-scoring

Stage-2 operates on the top-100 candidates retrieved per spectrum by Stage-1, re-ranking them using the InstaNovo decoder under teacher forcing, as detailed in Section 3.8. Table 4.5 and Figure 4.5 report Recall@k before and after re-scoring.

Table 4.5: Stage-1 retrieval versus Stage-2 neural re-scoring on the yeast test split (alignment and uniformity model, seed 42, $n = 111{,}312$ spectra). Re-scoring re-ranks the top-100 retrieved candidates per spectrum with the InstaNovo teacher-forced geometric-mean score. Δ denotes the absolute improvement in percentage points. Recall@100 is unchanged because re-scoring only re-orders the existing top-100 shortlist.

Recall@k	Stage 1 (Retrieval)	Stage 2 (Re-scored)	Δ
Recall@1	59.99%	**82.87%**	+22.88 pts
Recall@5	78.87%	**91.00%**	+12.13 pts
Recall@10	82.80%	**91.48%**	+8.68 pts
Recall@50	89.58%	**91.78%**	+2.20 pts
Recall@100	91.86%	91.86%	+0.00 pts

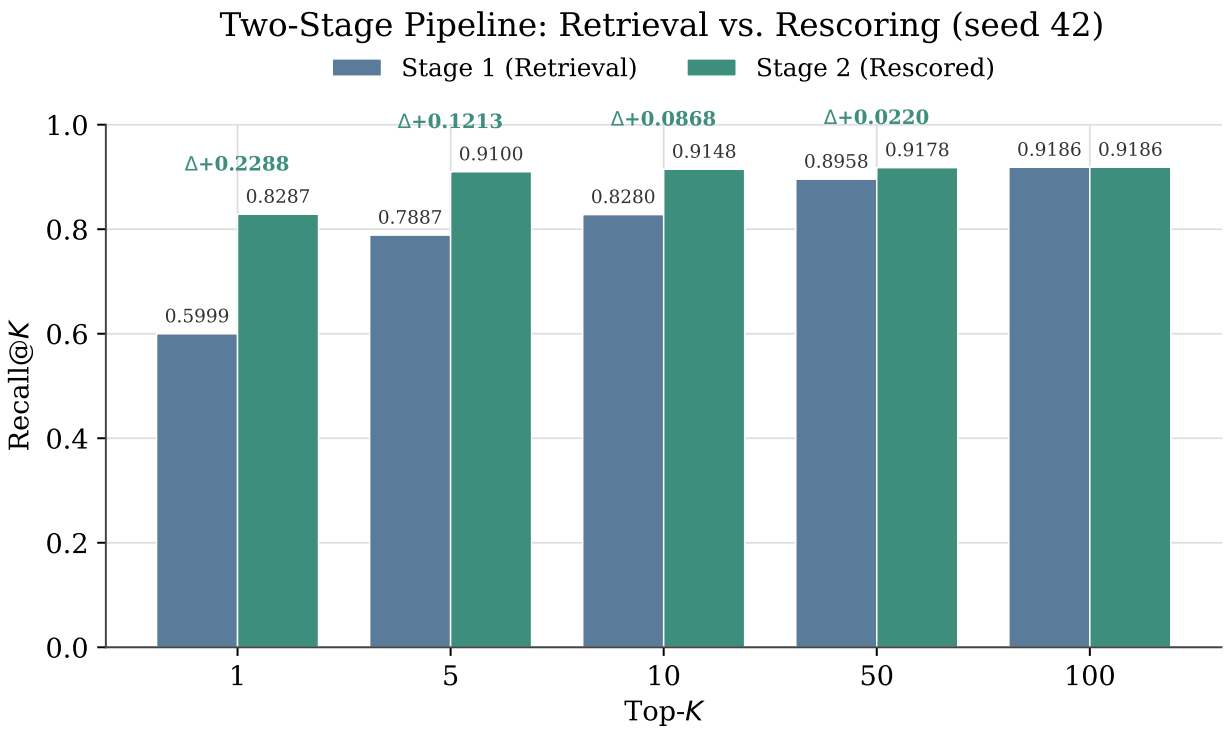


Figure 4.5: Recall@k values for Stage-1 retrieval and Stage-2 re-scoring on the yeast test split.

The re-scoring stage lifts Recall@1 by 22.88 percentage points, from 60.0% to 82.9%, confirming that the InstaNovo decoder carries fragmentation-level evidence that the cosine similarity score cannot capture. The diminishing gains at higher k are expected: re-scoring re-orders within the retrieved set and cannot recover candidates absent from the top-100 shortlist.

4.3.2 Retrieval Failure Analysis

The $9{,}056$ spectra not retrieved within the top-100 define the pipeline's ceiling. A breakdown by peptide length and charge state (Figure 4.6) shows that longer sequences are consistently harder to retrieve, and peptides with very rare charge states have zero recall, attributable to class imbalance in the training data. The cosine similarity distribution of retrieved PSMs is concentrated near 1, while non-retrieved PSMs show a broad, low-similarity distribution.

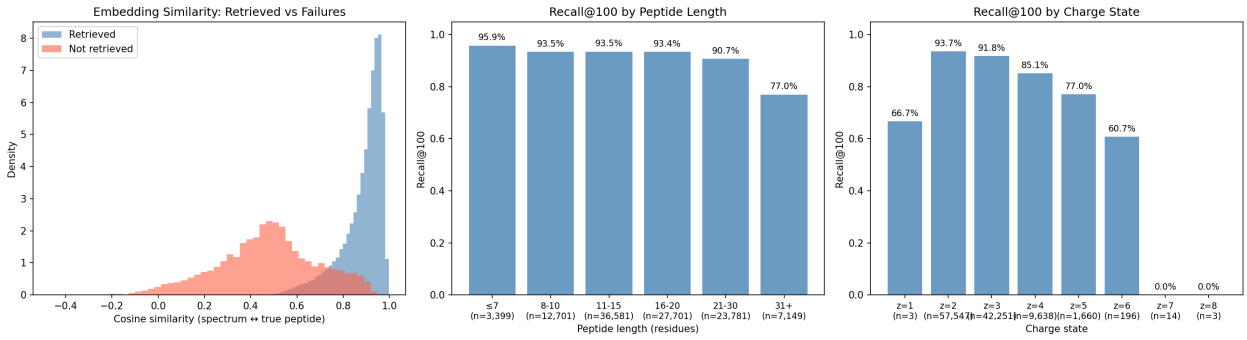


Figure 4.6: *Left*: cosine similarity score histograms for retrieved and non-retrieved PSMs, mean 0.47 ± 0.31. *Centre*: Recall@100 by peptide length. *Right*: Recall@100 by charge state.

To characterise the non-retrieved spectra independently, each was passed through the InstaNovo decoder with knapsack beam-search (5 beams). InstaNovo also failed to generate the correct sequence for these spectra, supporting the interpretation that they are genuinely difficult cases rather than retrieval artefacts. The failure taxonomy (Figure 4.7) shows that 57% of failures are length mismatches, 30.5% are empty predictions (no beam satisfies the mass constraint), 12% are wrong sequences, and 0.5% are partial matches. Failures are concentrated in the longest peptide length buckets.

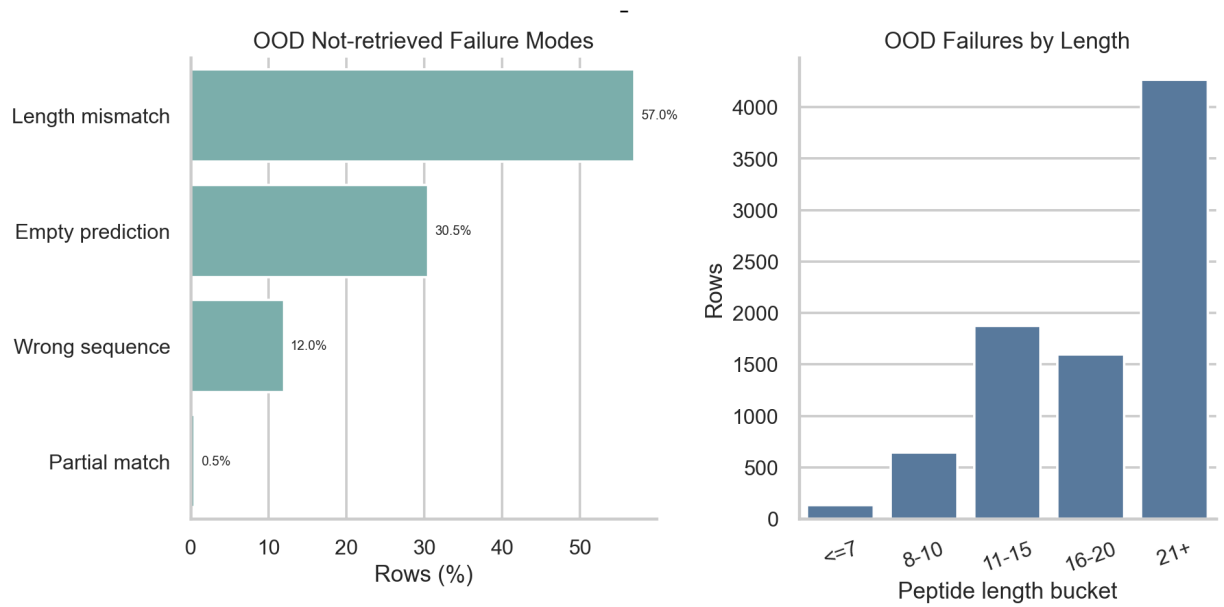


Figure 4.7: *Left*: percentage of non-retrieved spectra falling into each failure mode. *Right*: retrieval failures categorised by peptide length.

4.4 Cross-Proteome Generalisation: *Scalindua brodae*

To assess generalisation beyond the nine-species training distribution and to stress-test the pipeline at a larger database scale, INSearch was evaluated against the *Scalindua brodae* proteome. The protein file was digested in silico by a tryptic digestion script; a corresponding reversed-protein digest produced decoy sequences. Both target and decoy peptides were inserted into a single HNSW index, so that every

query spectrum competes against a large population of plausible but incorrect sequences, making the search substantially harder and more realistic than the yeast species benchmark.

The raw target and decoy sequences number $324{,}103$; after filtering for duplicates, there are $296{,}733$ target and $294{,}548$ decoy sequences. A total of $591{,}281$ peptides were encoded into the vector database in $16.60\,\mathrm{s}$ at a throughput of $35{,}611$ peptides/s on an NVIDIA A100 GPU with $40\,\mathrm{GB}$ of vRAM. The benchmark is evaluated on $9{,}068$ query PSMs: confident matches from a reference database search (Percolator $q \leq 0.01$) joined to their fragmentation spectra, with the reported peptide taken as ground truth.

Figure 4.8 shows two-stage Recall@K on this benchmark. Stage-1 Recall@100 reaches 55.5%, indicating that roughly 45% of the correct peptides are never placed in the candidate shortlist. Because re-scoring can only re-order the retrieved list, this ceiling bounds the achievable end-to-end accuracy regardless of scorer quality, and Recall@1 after re-scoring (55.2%) approaches that Stage-1 ceiling.

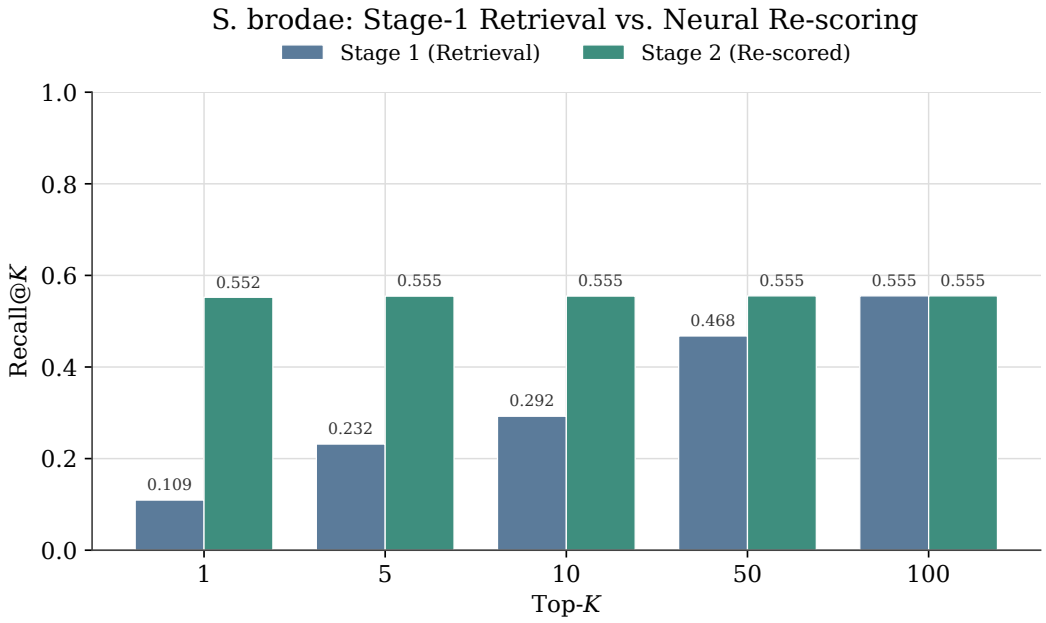


Figure 4.8: Two-stage Recall@K on the *S. brodae* benchmark. A total of $9{,}068$ query spectra are searched against a combined target+decoy database of $591{,}281$ peptides ($296{,}733$ targets + $294{,}548$ decoys). *Stage 1 (Retrieval)* is the HNSW approximate-nearest-neighbour recall; *Stage 2 (Re-scored)* is the recall after the InstaNovo decoder re-scores the full top-100 retrieved candidates per spectrum under teacher forcing.

Table 4.6: Stage-1 retrieval and neural re-scoring on *S. brodae*. Stage-1 Recall@K asks whether the true peptide appears anywhere in the top-K embedding candidates. Re-scored Recall@K is computed after re-scoring the stage-1 top-100 candidate set.

K	Stage-1 Retrieval Recall@K	Re-scored Recall@K
1	0.109	0.552
5	0.232	0.555
10	0.292	0.555
50	0.468	0.555
100	0.555	0.555

4.4.1 FDR and Target–Decoy Results

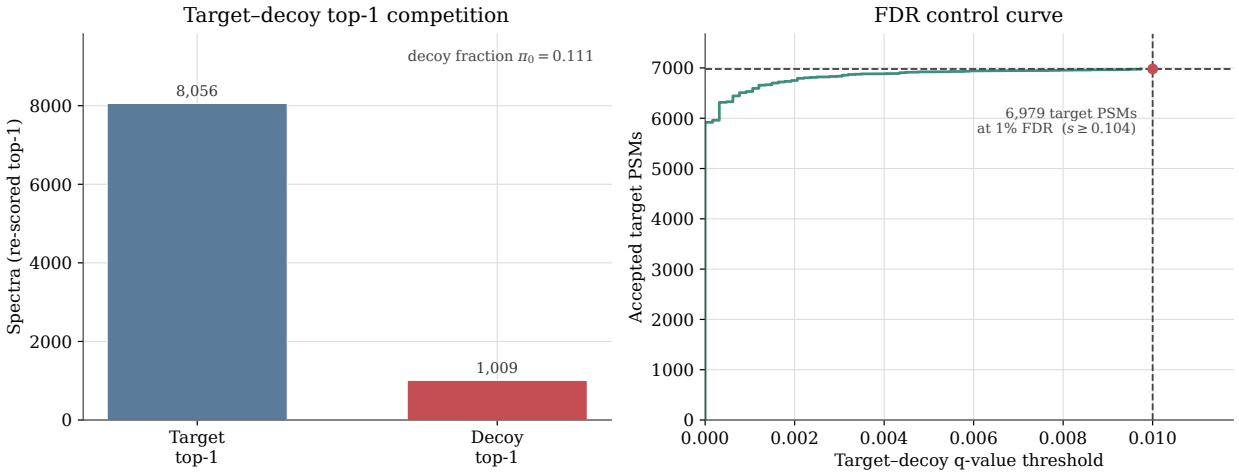


Figure 4.9: Target–decoy FDR control after neural re-scoring on the *S. brodae* benchmark. *Left:* re-scored top-1 outcome of the target–decoy competition: $8{,}056$ query spectra select a target peptide and $1{,}009$ select a decoy, a top-1 decoy fraction of $\pi_0 = 0.111$. *Right:* number of accepted target PSMs as a function of the target–decoy q-value threshold; at the 1% FDR threshold $6{,}979$ target PSMs are accepted, corresponding to a re-scored score cutoff of $s \geq 0.104$, where $s = e^{\overline{\log p}}$ is the InstaNovo teacher-forcing PSM score.

Because this benchmark has known ground-truth peptide labels, sequence-level recall is the primary measure of identification performance. Target–decoy FDR is reported only as a secondary calibration/yield diagnostic. In particular, accepted target PSMs are not necessarily correct ground-truth identifications: incorrect target peptides can also pass the FDR threshold if they score above decoys. Thus, the $6{,}979$ target PSMs accepted at 1% FDR indicate that the neural score separates targets from decoys under the target–decoy model, but they should not be interpreted as $6{,}979$ correct peptide identifications.

4.4.2 ANN Index: Build, Latency, and Recall Fidelity

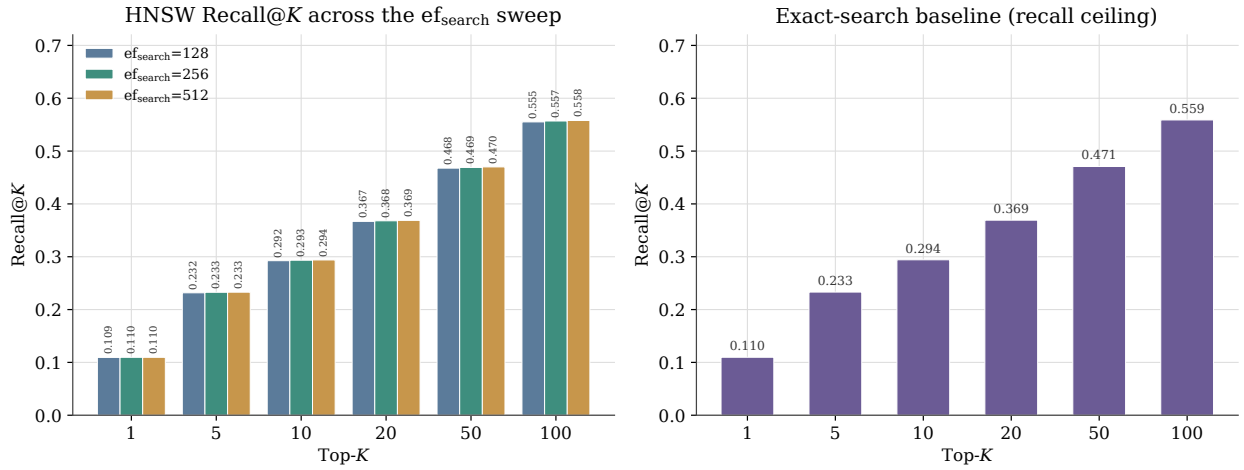


Figure 4.10: HNSW `ef_search` sweep on the *S. brodae* search space, over all retrieval depths $K \in \{1, 5, 10, 20, 50, 100\}$. The index stores all $591{,}281$ peptide embeddings (dimension 96) and is built with $M = 32$ and `ef_construction`$= 200$ in 35.3 s, on top of 16.6 s of embedding encoding (35.6k peptides/s), occupying 392.7 MB on disk. *Left:* Recall@K for `ef_search` $\in \{128, 256, 512\}$, at 0.036, 0.061 and 0.108 ms/query respectively; recall is essentially invariant to `ef_search` at this scale. *Right:* exact flat inner-product search baseline (1.41 ms/query), which is the recall ceiling over all K.

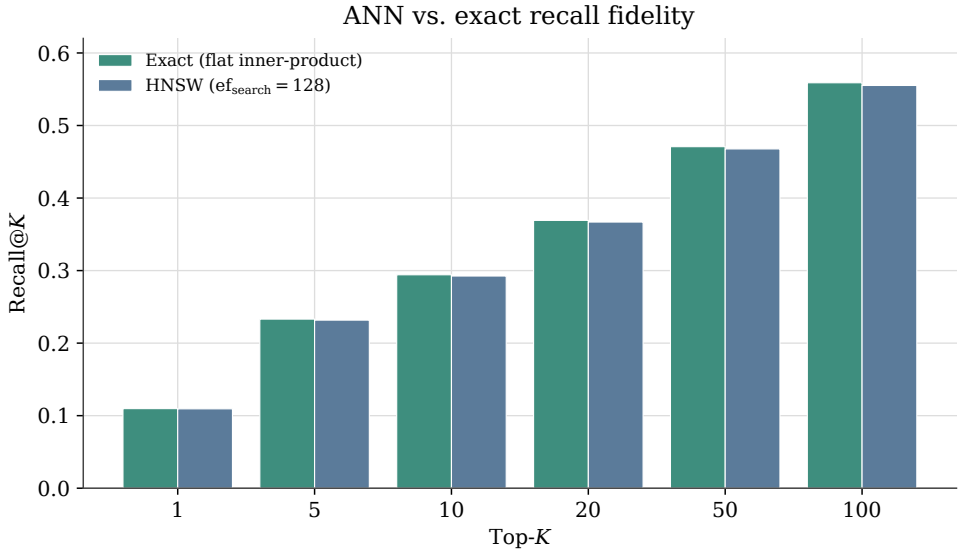


Figure 4.11: ANN versus exact recall fidelity on the *S. brodae* search space. Recall@K of the HNSW index (`ef_search`$= 128$) against the exact flat inner-product search across all $K \in \{1, 5, 10, 20, 50, 100\}$; the HNSW and exact recall differ by at most 0.004 at every K.

The HNSW index achieves up to $27{,}724$ queries/s at `ef_search`$= 128$, compared to 710 queries/s for exact search, a $39\times$ speedup with a recall penalty of at most 0.004 at every K. Recall is essentially invariant to `ef_search` at this database scale, confirming that the approximation is tight and that the

embedding quality rather than the index configuration is the limiting factor.

5. Conclusion, Limitations and Future Work

5.1 Conclusion

INSearch is a proposed AI-native framework for large-scale, mass spectrometry-based database search that aims to make peptide identification scale sub-linearly with database size. The pipeline has three components. The first and most critical component is the dual encoder: a spectrum encoder and a peptide encoder that jointly map two different modalities into a shared L_2-normalised latent space in which a matched spectrum-peptide pair is close under cosine similarity. The second is an approximate-nearest-neighbour retriever built on the Hierarchical Navigable Small World (HNSW) graph implemented in FAISS, which returns the top-k candidate peptides per query spectrum at $O(\log P)$ cost in the database size P. The third is a neural re-scorer that re-purposes the frozen InstaNovo decoder as a teacher-forced scoring function that assigns each candidate a final score based on the geometric mean of the per-residue log-probabilities and, accordingly, re-ranks the candidate shortlist.

The empirical study yielded three main findings. First, training the spectrum encoder from scratch on the available data induced severe dimensional collapse, whereas building it around the frozen, pre-trained InstaNovo backbone and training only a projection head removed the collapse and substantially raised Stage-1 recall on the held-out yeast split. Second, replacing the symmetric InfoNCE objective with a direct alignment-and-uniformity decomposition (augmented with a decoy margin) produced the best-aligned, least-collapsed embeddings and the strongest retrieval, reaching Recall@1/5/100 of $60.0/78.9/91.9\%$ on the yeast test split. Third, neural re-scoring of the retrieved shortlist sharply improved top-ranked accuracy, lifting yeast Recall@1 from 60.0% to 82.9%.

On the external *Scalindua brodae* proteome (a stricter, cross-proteome test against a combined target-decoy database of $591{,}281$ peptides), the same pipeline retrieved the correct peptide within the top 100 for 55.5% of the $9{,}068$ query spectra, and re-scoring raised Recall@1 to 55.2%, close to that Stage-1 ceiling. Under target-decoy competition with the neural score, $6{,}979$ target PSMs (77% of the queries) were accepted at a 1% false discovery rate. Together these results show that the dual-encoder-plus-re-scorer design is viable and fast, while also exposing a clear retrieval ceiling on out-of-distribution data that bounds the achievable end-to-end identification rate.

5.2 Limitations

The conclusions above are qualified by several limitations.

- **Out-of-distribution retrieval ceiling.** On the *S. brodae* proteome the Stage-1 Recall@100 is only 55.5%, meaning that for roughly 45% of spectra the correct peptide is never placed in the candidate shortlist. Because re-scoring can only re-order the retrieved list, this ceiling bounds the achievable end-to-end accuracy regardless of how strong the scorer is. The gap in performance between the yeast test split and the cross-proteome (S. brodae) indicates that the current small-scale model isn't sufficient for providing strong generalization ability and further support the need for training on a larger scale.
- **No direct baseline comparison.** The framework was not benchmarked head-to-head against established database-search engines (for example MSFragger or Comet) or against competing learned systems (Casanovo-DB, yHydra) on identical data and hardware. The scalability argument

therefore rests on the asymptotic $O(\log P)$ cost of HNSW rather than on a measured runtime-versus-database-size comparison against a linear baseline.

- **Cost of re-scoring.** While stage-1 retrieval is cheap, the second stage is expensive: the InstaNovo decoder must score each candidate peptide for each spectrum, so the cost scales approximately linearly with the candidate pool size K. This can dominate per-spectrum runtime and partially offset the speed-up gained from approximate nearest-neighbour retrieval. Therefore, a practical AI-native search system depends not only on high stage-1 Recall@100 but also on high stage-1 Recall@10 or Recall@20, so that only a small candidate pool needs neural re-scoring.
- **Scale of the evaluation.** Training used eight species, and evaluation was limited to one held-out species (yeast) and one external proteome (*S. brodae*); the largest index searched contained $\sim 6 \times 10^5$ peptides, well below the 10^7–10^9 scale at which the sub-linear advantage is most consequential.
- **Interpretability of the latent embedding space**.A proper method should be used to interpret the latent features captured by the model in the latent space. Currently, a simple dimensionality reduction technique is used (t-SNE projection by precursor m/z ratio). While effective, it doesn't provide a complete perspective on the interpretability of the latent space.

5.3 Future Work

Several directions follow directly from these limitations.

- **Direct baselines and scaling curves.** The most important next step is a head-to-head evaluation against MSFragger/Comet and against Casanovo-DB and yHydra on identical spectra, reporting both identification rate at 1% FDR and wall-clock runtime as the database size P is swept so that the sub-linear scaling claim is substantiated empirically rather than asymptotically.
- **Closing the out-of-distribution gap.** The Stage-1 ceiling motivates stronger contrastive training: explicit hard-negative mining, a MoCo-style (He et al., 2020) momentum queue to enlarge the effective negative pool beyond the batch size, and additional training species or augmentation to improve cross-proteome generalisation.
- **Fine-tuning the scorer for database search.** Following the open question raised by Casanovo-DB (Ananth et al., 2024), the InstaNovo decoder could be fine-tuned specifically for the teacher-forced scoring task rather than used purely as a frozen de novo model, which may further improve both ranking and FDR calibration.
- **A third refinement stage.** A diffusion-based sequence-refinement step, as used in recent InstaNovo variants, could be appended after re-scoring to recover near-miss candidates, particularly the long and heavily modified peptides that dominate the current failure cases.
- **Open-PTM search at scale.** Because modifications appear as small displacements in the joint embedding space rather than as combinatorial database expansion, the framework is naturally suited to open-modification search; demonstrating this at the 10^7–10^8-peptide scale would be the clearest validation of the design's central advantage.
- **Interpretability and Explainability Methods**. After the scaling phase, a search for suitable methods for interpretability and explainability must be conducted so as not to treat the model as a black-box oracle but rather as a trustworthy model that can be confidently deployed in downstream tasks.

- **Granular ablation of the loss components.** The loss-function comparison in this work evaluates the contrastive (InfoNCE plus variance) and alignment-and-uniformity configurations as bundles; a future ablation should isolate the individual contributions of the reversed-peptide decoy column and the decoy margin term, quantifying each component's effect on retrieval performance and FDR calibration.

Acknowledgements

I would like to express my gratitude and appreciation towards AIMS and Google DeepMind for their unwavering support in my postgraduate academic journey and for funding the experimentation costs.

I would like to extend my sincere thanks to my supervisors, Dr Konstantinos Kalogeropoulos at the Technical University of Denmark; Mr Jeroen Van Goey, the head of BioAI at InstaDeep; Jemma Daniel; and Divanisha Patel, who both work as research engineers at InstaDeep, for their guidance through the research phase and for their technical insights.

References

Aebersold, R. and Mann, M. Mass-spectrometric exploration of proteome structure and function. *Nature*, 537(7620):347–355, 2016.

Altenburg, T., Muth, T., and Renard, B. Y. yhydra: Deep learning enables an ultra fast open search by jointly embedding ms/ms spectra and peptides of mass spectrometry-based proteomics. *bioRxiv*, pages 2021–12, 2021.

Ananth, V., Sanders, J., Yilmaz, M., Wen, B., Oh, S., and Noble, W. S. A learned score function improves the power of mass spectrometry database search. *Bioinformatics*, 40(Supplement_1):i410–i417, 2024.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. URL https://arxiv.org/abs/2002.05709.

Craig, R. and Beavis, R. C. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20 (9):1466–1467, 2004.

Elias, J. E. and Gygi, S. P. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, 2007. doi: 10.1038/nmeth1019.

Eloff, K., Kalogeropoulos, K., Mabona, A., Morell, O., Catzel, R., Rivera-de Torre, E., Berg Jespersen, J., Williams, W., van Beljouw, S. P., Skwark, M. J., et al. Instanovo enables diffusion-powered de novo peptide sequencing in large-scale proteomics experiments. *Nature Machine Intelligence*, 7(4): 565–579, 2025.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019. URL http://arxiv.org/abs/1911.05722.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning, 2020. URL https://arxiv.org/abs/1911.05722.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. *CoRR*, abs/2110.09348, 2021. URL https://arxiv.org/abs/2110.09348.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

Kalogeropoulos, K., Van Goey, J., Jenkins, T. P., and Eloff, K. M. A framework for database search with ai models in mass spectrometry-based proteomics. *Journal of Proteome Research*, 25(5):2234–2242, 2026.

Keich, U. and Noble, W. S. On the importance of well-calibrated scores for identifying shotgun proteomics spectra. *Journal of proteome research*, 14(2):1147–1160, 2015.

Kim, S. and Pevzner, P. A. Ms-gf+ makes progress towards a universal database search tool for proteomics. *Nature communications*, 5(1):5277, 2014.

Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., and Nesvizhskii, A. I. Msfragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature methods*, 14(5):513–520, 2017.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Malkov, Y. A. and Yashunin, D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Perkins, D. N., Pappin, D. J., Creasy, D. M., and Cottrell, J. S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *ELECTROPHORESIS: An International Journal*, 20(18):3551–3567, 1999.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. *2007 15th European Signal Processing Conference*, pages 606–610, 2007. URL https://api.semanticscholar.org/CorpusID:12184201.

Tran, N. H., Zhang, X., Xin, L., Shan, B., and Li, M. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017. doi: 10.1073/pnas.1705691114.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Venable, J. D. and Yates, J. R. Impact of ion trap tandem mass spectra variability on the identification of peptides. *Analytical chemistry*, 76(10):2928–2937, 2004.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 2020.

Xiong, P., Xu, H., and Zheng, H. Supervised contrastive learning leads to more reasonable spectral embeddings. *Analytical Chemistry*, 97(37):20137–20146, 2025.

Yilmaz, M., Fondrie, W. E., Bittremieux, W., Melendez, C. F., Nelson, R., Ananth, V., Oh, S., and Noble, W. S. Sequence-to-sequence translation from mass spectra to peptides with a transformer model. *Nature communications*, 15(1):6427, 2024.

Zolg, D. P., Wilhelm, M., Schmidt, T., Médard, G., Zerweck, J., Knaute, T., Wenschuh, H., Reimer, U., Schnatbaum, K., and Kuster, B. Proteometools: Systematic characterization of 21 post-translational protein modifications by liquid chromatography tandem mass spectrometry (lc-ms/ms) using synthetic peptides. *Molecular & Cellular Proteomics*, 17(9):1850–1863, 2018.

Appendix A. Supplementary Results

A.1 Spectrum–Peptide Similarity Matrix

Figure A.1 shows the similarity matrix for a randomly selected batch from the test dataset, containing 128 PSMs. Each entry represents the cosine similarity between a spectrum and a peptide sequence in the batch, where brighter entries indicate higher similarity. The diagonal entries correspond to the similarity between each spectrum and its ground truth peptide sequence, and the consistently bright diagonal confirms that the encoder reliably assigns high similarity to correct spectrum–peptide pairs.

However, several off-diagonal entries also appear bright, indicating high similarity between a spectrum and a peptide other than its ground truth. This can arise from two sources. First, duplicate PSMs within the batch (where multiple spectra map to the same peptide sequence) will naturally produce high off-diagonal similarity scores, as the encoder correctly embeds those spectra close to their shared ground truth. Second, and more fundamentally, some peptides share sufficient physicochemical properties with the ground truth sequence that they produce spectrally similar fragmentation patterns, causing the encoder to map them to proximal regions of the embedding space. Both phenomena are expected and do not necessarily reflect encoder failure; rather, they reflect the inherent ambiguity in the spectrum–peptide matching problem and motivate the need for fine-grained discrimination beyond cosine similarity alone.

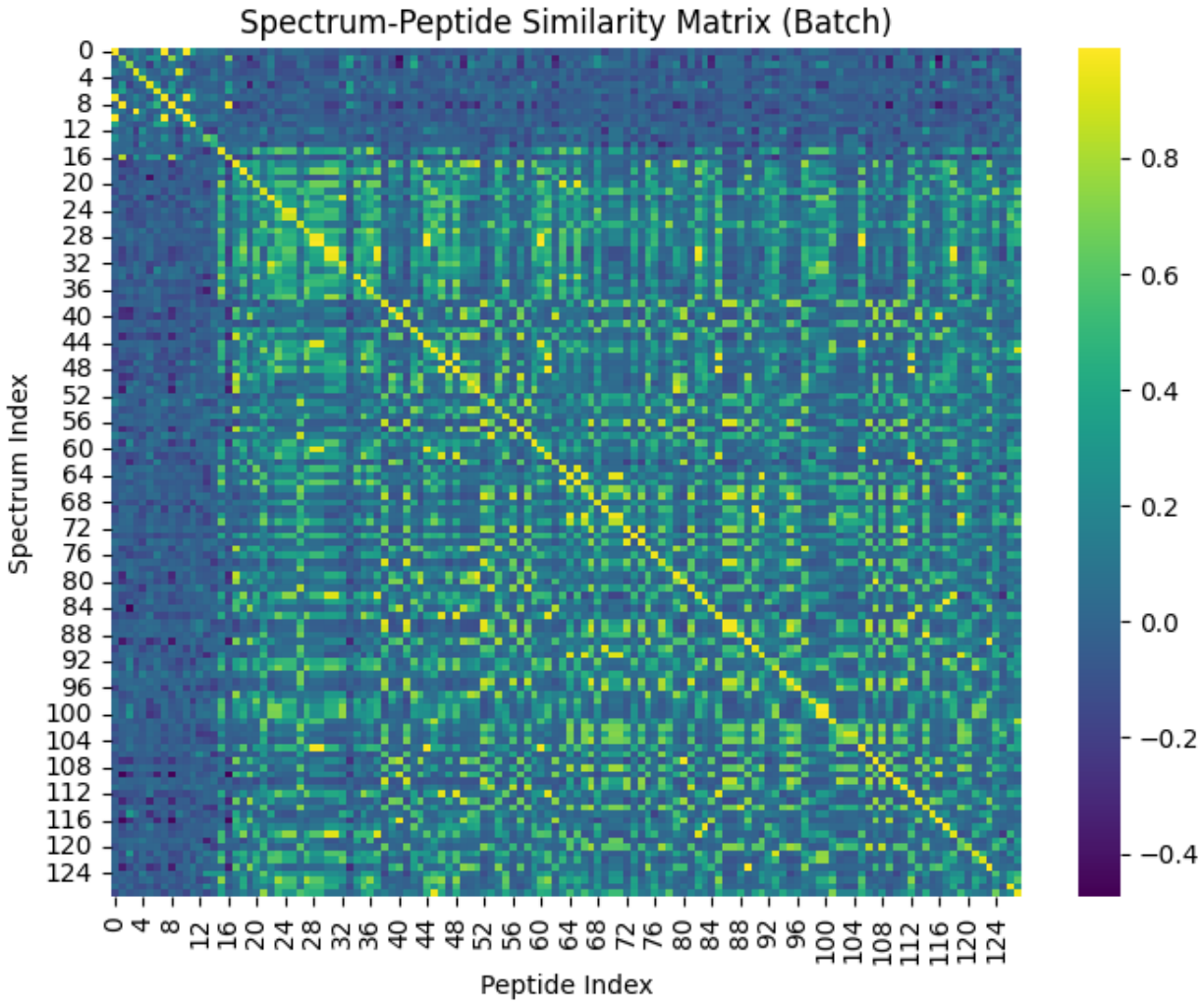


Figure A.1: Similarity matrix for a random batch selected from the test set, consisting of 128 PSMs.

A.2 Embedding-Space Collapse Diagnostics

To verify that the dual-encoder does not suffer from representational collapse (a failure mode in which the encoder maps all inputs to a degenerate, low-variance region of the embedding space), a comprehensive set of diagnostic metrics was computed over the full yeast test set of 111,312 embeddings with a dimensionality of $d = 96$.

A.2.1 Feature Variance

The per-feature standard deviation of the embeddings serves as a first-order collapse indicator. For a well-distributed embedding space of dimensionality d, the expected per-feature standard deviation of unit-normalised vectors is $1/\sqrt{d}$. For $d = 96$, this theoretical target is $1/\sqrt{96} \approx 0.1021$. The observed values were:

Table A.1: Per-feature standard deviation of encoder embeddings.

Encoder	Observed Std Dev	Target ($1/\sqrt{96}$)
Spectrum	0.1015	0.1021
Peptide	0.1017	0.1021

Both encoders produce per-feature standard deviations within 0.6% of the theoretical target, indicating that the embedding dimensions are utilised uniformly and that no dimensional collapse has occurred.

A.2.2 Within-Encoder Isotropy

The mean off-diagonal cosine similarity within each encoder's embedding set measures isotropy: the degree to which embeddings are distributed without systematic bias toward any particular direction. Values close to zero indicate a well-spread, isotropic distribution:

Table A.2: Mean off-diagonal cosine similarity within each encoder.

Encoder	Mean Off-Diag Cosine Sim	Ideal
Spectrum vs Spectrum	0.0110	≈ 0.0
Peptide vs Peptide	0.0072	≈ 0.0

Both values are close to zero, confirming that the embeddings are not systematically aligned with one another and that the encoder has not collapsed to a low-dimensional subspace.

A.2.3 Cross-Encoder Alignment and Discrimination

The retrieval-relevant diagnostic measures the cosine similarity between matched spectrum–peptide pairs (positive pairs) and unmatched pairs (negative pairs):

Table A.3: Cross-encoder cosine similarity between positive and negative pairs.

Metric	Observed	Ideal
Mean positive-pair cosine similarity	0.8514	$\rightarrow 1.0$
Mean negative-pair cosine similarity	0.0083	≈ 0.0
Gap (positive − negative)	0.8431	larger is better

The large gap of 0.8431 between positive and negative pair similarities confirms strong cross-modal alignment and effective discrimination between matched and unmatched pairs. The alignment loss of $\mathcal{L}_{\text{align}} = 0.2972$ (Equation 2.2.2), while not yet approaching the ideal of zero, reflects the inherent difficulty of the task and is consistent with the retrieval performance reported in Section 4.2.3.

A.2.4 Effective Rank and Uniformity

The effective rank of the embedding matrix measures the intrinsic dimensionality of the learned representations: the number of dimensions that carry meaningful variance. A higher effective rank indicates better utilisation of the available embedding space:

Table A.4: Effective rank and uniformity loss of encoder embeddings.

Encoder	Effective Rank	Maximum	Uniformity Loss
Spectrum	58.3	96	-3.6948
Peptide	58.3	96	-3.7010

Both encoders achieve an effective rank of 58.3 out of a maximum of 96, indicating that approximately 60% of the available embedding dimensions carry meaningful variance. While full utilisation of all 96 dimensions would be ideal, an effective rank of 58.3 is consistent with healthy, non-collapsed representations and suggests room for improvement through scaling or additional regularisation. The uniformity loss values of -3.69 and -3.70 (Equation 2.2.3) are negative as expected, indicating that embeddings are spread across the hypersphere rather than collapsed to a narrow region.

A.2.5 Collapse Assessment

Based on the diagnostics above, both the spectrum and peptide encoders are assessed as **healthy**, exhibiting no signs of representational collapse. The embedding space is well-utilised, isotropic, and discriminative, providing a sound foundation for the retrieval pipeline described in this work.