

REFLECTION PRETRAINING ENABLES TOKEN-LEVEL ERROR CORRECTION IN BIOLOGICAL SEQUENCE GENERATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Large language models with natural language (e.g. English) have shown that generating **auxiliary tokens**—intermediate outputs not part of the final answer—enables powerful capabilities such as error correction, self-reflection, and more reliable reasoning. Methods like **Chain-of-Thought prompting** exploit the high expressiveness of natural languages (e.g., English), allowing models to verbalize internal states and perform complicated reasoning over text space. In contrast, **biological sequence models** (e.g., proteins, RNA, DNA) operate over token spaces with limited expressiveness, restricted to amino acids tokens or nucleotides tokens. As a result, these models lack mechanisms for externalized reasoning and are confined to producing **only** final sequence tokens without self-correction. In this work, we introduce **Bio-reflection pretraining**, a new framework that augments biological sequence models with an auxiliary `<reflect>` token. We select the reflection token because it provides flexible way for token-level modifications—such as error flagging, correction, swapping, and deletion—that directly target the types of mistakes most common and most urgent in biological sequence generation. By injecting synthetic errors during training and requiring the model to explicitly mark and correct them, we teach the model to engage in reflection and self-error-correction. This approach increases the effective expressiveness of biological sequence languages, enabling intermediate reasoning steps previously unattainable in this domain. We evaluate our method on the challenging task of *de novo peptide sequencing*, where intermediate reasoning is critical and the ground-truth label is unique and clearly defined. We demonstrate both theoretically and empirically that reflection pretraining substantially improves model accuracy on this task and enhances robustness against overfitting. Beyond accuracy gains, our framework enables **human-in-the-loop interaction**, allowing experts to guide or override reflection points during sequence generation. Taken together, reflection pretraining offers a principled path toward more interpretable and steerable biological sequence models, narrowing the gap between natural language models and their biological counterparts. **Codes, model outputs and model weights are available at *Anonymous GitHub repository* for full reproducibility.**

1 INTRODUCTION

Deep learning (LeCun et al., 2015) has significantly advanced the field of biology, with an increasing number of neural models being trained to generate and predict biological sequences such as DNA (Alipanahi et al., 2015; Zhang et al., 2021), RNA (Yang et al., 2022; Deng et al., 2022), and proteins (Xiao et al., 2025; Rives et al., 2021; Rao et al., 2021; Jumper et al., 2021; Lin et al., 2023). However, current biological sequence-generation models are constrained to produce only **answer tokens** directly related to specific tasks (e.g., drug design, de novo sequencing). This generation paradigm mirrors **conventional** natural language processing models (Naveed et al., 2023) before LLMs, where outputs are limited to final answers without intermediate reasoning or deliberation.¹ Recent work (Wei et al., 2022; Zhang et al., 2024a) has demonstrated that this answer-only generation

¹For example, a machine translation model outputs just “goodbye” for the input “au revoir,” or a mathematical language model returns just answer “121” for the input “11 × 11,” without revealing intermediate steps.

054 approach is suboptimal, both in terms of theoretical expressiveness and empirical performance. While
 055 a full theoretical analysis is provided in the Appendix, the intuition is straightforward: solving
 056 complex tasks often involves trial-and-error, exploration and even initial incorrect outputs before
 057 arriving at a final solution. Models constrained to generate only final answers are fundamentally
 058 incapable of performing this kind of structured, exploratory computation and thus fail to handle
 059 complex solution discovery effectively.

060 Chain-of-Thought (CoT) (Wei et al., 2022) prompting fundamentally changes how answers are
 061 generated in natural language models. Traditional neural models directly map an input sequence to a
 062 sequence of **answer tokens**, expressed as: $x_i : x_n \Rightarrow \langle \text{answer}_1 \rangle \langle \text{answer}_2 \rangle \cdots \langle \text{answer}_m \rangle$.

063 In contrast, CoT introduces interleaved **non-answer tokens** that enable intermediate reasoning (Wei
 064 et al., 2022):

$$065 \quad x_i : x_n \Rightarrow \langle \text{answer}_1 \rangle [\text{non-answer}_1] [\text{non-answer}_2] \langle \text{answer}_2 \rangle \cdots \langle \text{answer}_m \rangle.$$

066
 067 Although these non-answer tokens, $[\text{non-answer}]$, are discarded in the final output, they signif-
 068 icantly enhance the model’s capabilities by enabling it to *perform iterative computation, correct*
 069 *earlier errors, and reason across multiple steps* before reaching solutions.

070 This augmentation enables natural language (e.g. English) models to perform human-like reasoning
 071 such as self-correcting. Under suitable assumptions, CoT-enhanced models can even approach the
 072 theoretical upper bound of Turing completeness (Li et al., 2024) (detailed in Appendix), a level of
 073 expressiveness unattainable by direct-answer generation alone (Delétang et al., 2022).

074 However, such a reasoning process is only seen in LLMs due to the expressive capacity of human
 075 natural languages (e.g., English). Highly expressive languages can encode rich, structured information
 076 from hidden states \mathbf{h} into tokens, including reflections on prior errors, intermediate computations,
 077 and procedural knowledge. In contrast, language like protein (Rives et al., 2021; Yilmaz et al.,
 078 2022) or RNA (Yang et al., 2022) relies on token and grammar systems with limited expressiveness,
 079 restricting models to **output only the next amino acid or nucleotides**. **In this paper, we formally**
 080 **demonstrate that the expressiveness of the foundational language directly affects the theoretical**
 081 **upper bound on the model’s overall reasoning capability**.

082 In this work, we augment biological sequence models with *non-answer reflection tokens* through
 083 a novel **reflection-based pretraining** paradigm. We introduce a dedicated $\langle \text{reflect} \rangle$ token,
 084 chosen because it allows token-level modifications—such as error flagging, correction, swapping, and
 085 deletion—which directly address the types of mistakes most common and most critical in biological
 086 sequence generation. Unlike conventional pretraining, which limits models to only predicting
 087 sequence answer tokens, our approach explicitly equips models with the ability to generate reflection
 088 outputs that capture error-corrective signals. We focus on the challenging task of *de novo* protein
 089 sequencing, where the space of possible outputs is vast, yet the ground-truth label is *unique* and
 090 *well defined*—making it an ideal benchmark for testing reasoning-based biological approaches. We
 091 construct a large-scale and carefully curated reflection dataset, specifically designed to simulate
 092 error-prone predictions. Within this framework, we introduce two complementary mechanisms for
 093 error injection, enabling the model to encounter realistic mistakes and learn to *self-reflect* on its
 094 predictions. By training the model to detect, annotate, and revise these errors through reflection
 095 tokens, we show that biological sequence models can acquire a structured process of *self-correction*,
 096 thereby improving both accuracy and interpretability compared to standard pretraining.

097 **Our extensive experiments yield the following key findings:** (1) With appropriate training tech-
 098 niques and data augmentation, protein sequence models can perform **self-correction and self-**
 099 **reflection** capabilities that go beyond simply generating just answer tokens; (2) Finetuning alone
 100 does not confer reflection ability in bio-models; such capacity **emerges only through pretraining**;
 101 (3) Within a certain threshold, increasing the number of manually injected errors during training
 102 improves the model’s ability to **self-correct**; (4) Reflection pretraining leads to significant accuracy
 103 gains, driven by both **enhanced generational ability** and a **counter-overfitting** effect during training
 104 and error-data augmentation; (5) The resulting models enable **human-in-the-loop generation**, al-
 105 lowing experts to actively engage with reflection and error-correction steps. This interactivity brings
 106 biological language models closer in capability and usability to large natural language models.
 107

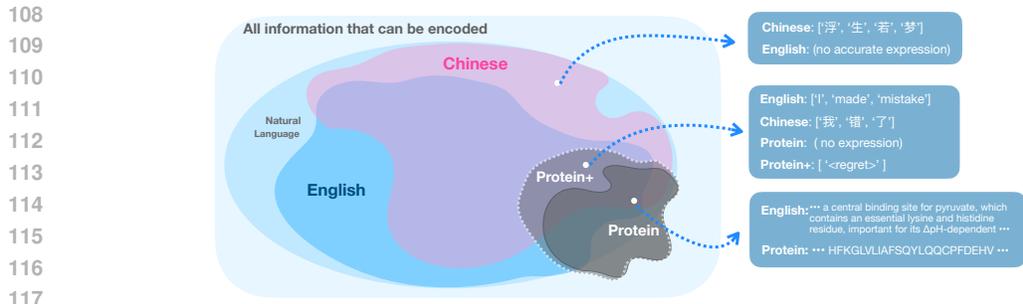


Figure 1: Expressiveness of Languages.

2 BIO LANGUAGE EXPRESSIVENESS

2.1 REFLECTION, ERROR-CORRECTION, AND THE LIMITS OF BIOLOGICAL TOKENS

Neural network expressiveness depends strongly on the tokens a model can output. Biological sequence models generate only **answer tokens** (amino acids or nucleotides), forcing them to commit to final sequences without intermediate reasoning. This prevents (1) multi-step deliberation, (2) flexible computation, and (3) use of externalized memory.

Natural language models overcome these limitations using **auxiliary reasoning tokens** (e.g., Chain-of-Thought), which allow intermediate thoughts, checks, and revisions. Natural languages are expressive enough to encode logic, heuristics, and self-evaluation.

Biological languages lack such expressive capacity: amino-acid alphabets were shaped for biochemical function, not reasoning. They cannot express concepts like “this residue is incorrect” or “swap these two positions.”

To address this, we introduce the notion of **biological language expressiveness** and augment biological models with an auxiliary token, `<reflect>`. Unlike amino-acid tokens, `<reflect>` encodes higher-level information—flagging likely mistakes and suggesting corrections, swaps, or deletions. These local edits are crucial because small token-level errors can significantly disrupt biological function. The reflection token provides a controlled mechanism for identifying and correcting such errors.

2.2 LANGUAGE EXPRESSIVENESS

We formalize a language as $\mathbf{L} = (\mathcal{G}, \mathcal{V})$, where \mathcal{G} is its grammar and \mathcal{V} its vocabulary. The set of meanings representable in the language is $\mathbb{S}_{\mathbf{L}}$. The **expressiveness** of \mathbf{L} is:

$$\text{Expressiveness}(\mathbf{L}) := |\mathbb{S}_{\mathbf{L}}|.$$

Even with infinitely many valid sequences, languages differ in the meanings they can encode. If $\mathbb{S}_{\mathbf{L}_1} \subset \mathbb{S}_{\mathbf{L}_2}$, then \mathbf{L}_2 is strictly more expressive.

Natural languages (e.g., English, Chinese) are highly expressive due to their large vocabularies and flexible grammar, enabling complex reasoning, explanation, and abstraction.

Biological languages such as protein sequences, $\mathbf{L}_{\text{protein}} = (\mathcal{G}_{\text{protein}}, \mathcal{V}_{\text{protein}})$ with 20 amino acids, are excellent for encoding structure and function but extremely limited for reasoning. Statements like “I am wrong” have no analogue in $\mathbb{S}_{\text{protein}}$.

2.3 LANGUAGE EXPRESSIVENESS DETERMINES CoT EXPRESSIVENESS

Biological sequence models generate only answer tokens, so any internal signals—uncertainty, regret, or error detection—remain hidden and never appear in the output (Fig. 6). Thus, these models cannot externalize reasoning or perform self-correction.

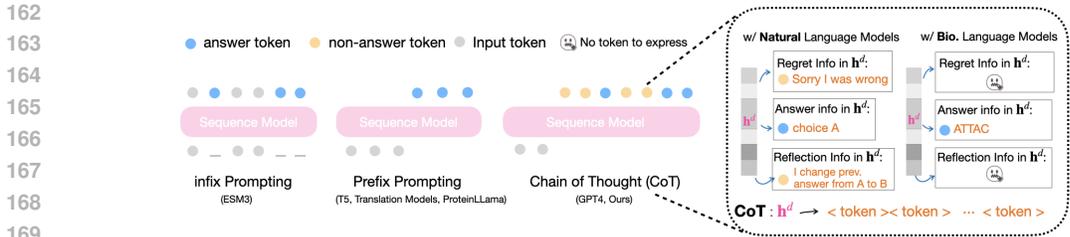


Figure 2: Comparative framework for prompting in natural and biological language models.

In natural language models, Chain-of-Thought introduces **non-answer tokens** that serve as intermediate reasoning steps (Fig. 6). These allow the model to revise or expand its reasoning before committing to an answer.

Essence of CoT

CoT maps hidden states \mathbf{h} into language tokens, $\mathbb{S}_\Theta \rightarrow \mathbb{S}_L$, enabling signals such as “I made a mistake” to appear as non-answer tokens before final answers.

Because amino-acid or nucleotide alphabets cannot encode such concepts, the CoT expressiveness of biological models is effectively **zero**. The `<reflect>` token provides the necessary channel to express intermediate reasoning and perform explicit error correction.

3 REFLECTION PRETRAINING FOR PROTEIN (PEPTIDE) SEQUENCE PREDICTIONS

3.1 TASK CHOICE AND PROBLEM FORMULATION

To this end, we propose *reflection-pretraining* for protein sequencing models to allow CoT reasoning beyond generating solely answer tokens (amino acids). We choose the the task of **de novo peptide sequencing** (Fig. 3) for five key reasons: 1) It is a foundational problem in protein sequence prediction, as peptide sequencing remains the primary method for determining amino acid sequences from nature. 2) The task is reasoning-intensive, requiring the model to perform complex computations on spectrum signals-well aligned with the reflection paradigm. 3) Large-scale training data is readily available. 4) Evaluation is precise, with each input spectrum paired with a gold-standard sequence, unlike protein language modeling with ambiguous metrics. 5) Each token corresponds to a discrete reasoning step, enabling fine-grained, token-level reflection.

In the *de novo* peptide sequencing task (Fig. 3), the model is given a spectrum instance $\mathbf{H} = \{\mathbf{I}, c, m\}$, produced by a mass spectrometer (Fig. 3 when provided with a biological sample (Fig. 3. The spectrum consists of:

- $\mathbf{I} = \{(m/z_1, i_1), (m/z_2, i_2), \dots, (m/z_k, i_k)\}$, a set of k observed mass-to-charge and intensity pairs, filtered by a signal threshold, (Fig. 3 MS/MS)
- $c \in \mathbb{Z}^+$, the charge state of the precursor ion, and
- $m \in \mathbb{R}^+$, the total measured mass of the peptide.

The objective is to **predict the underlying amino acid (protein) sequence** $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$, where each token $a_i \in \mathcal{V}_{\text{protein}} = \{20 \text{ amino acids tokens}\}$ belongs to the vocabulary of the protein language $\mathcal{L}_{\text{protein}} = (\mathcal{G}_{\text{protein}}, \mathcal{V}_{\text{protein}})$. The mapping $\mathbf{H} \mapsto \mathbf{A}$ requires the model to reason over the structure and intensity patterns in \mathbf{I} while conforming to biochemical constraints imposed by c and m .

Note that, although we describe our method in the context of the protein sequence space, it applies to other biological sequence prediction tasks-including RNA, DNA, and synthetic polymers.

216
217
218
219
220
221
222

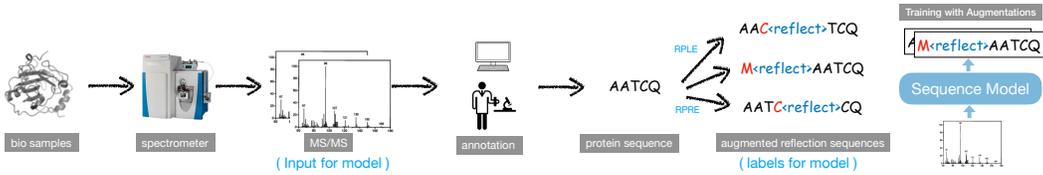


Figure 3: De novo peptide sequencing workflow using tandem mass spectrometry.

223
224
225

3.2 REFLECTION TOKEN AUGMENTED PROTEIN SEQUENCE FOR NEXT TOKEN PREDICTION TRAINING

229
230
231
232
233
234
235

One of the most powerful forms of reasoning is self-reflection and error correction. Prior work shows that sequence models often encode a sense of “regret” in their latent state \mathbf{h} -that is, the realization of an error *after* generating an answer token $\langle a \rangle$. In large language models (LLMs), the high expressiveness of the English language $\mathcal{L}_{\text{english}}$ enables this information to be verbalized, making reflection a highly effective mechanism for improving reasoning accuracy. However, in protein models, this is not possible: the latent regret state $\mathbf{h}_{\text{regret}}$ cannot be decoded into any token $a \in \mathcal{V}_{\text{protein}}$, i.e., $\mathbf{h}_{\text{regret}} \Rightarrow a$ is infeasible.

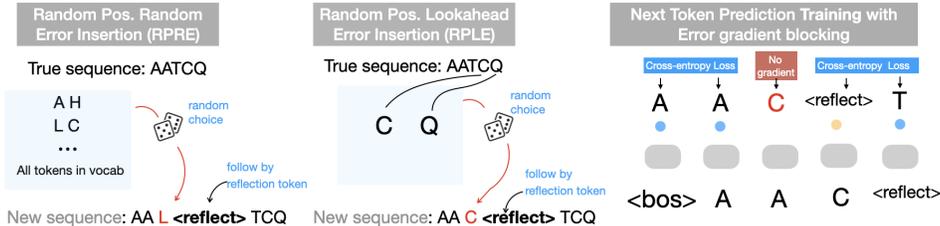
236
237
238
239

To enable self-reflection in protein sequence models, we augment the vocabulary with a reflection token, defining $\mathcal{V}_{\text{protein}^+} = \mathcal{V}_{\text{protein}} \oplus \langle \text{reflect} \rangle$. We then construct a modified training dataset $\mathcal{D}_{\text{de novo}^+}$ (Figure 3) by injecting synthetic errors (e.g., incorrect amino acids) into sequences and appending corrections using $\langle \text{reflect} \rangle$.

240
241
242
243

In this section, we introduce two novel strategies for error injection (Fig. 4), as well as our dynamic data updating and gradient blocking strategy for training protein models with reflection-based next-token prediction on the *de novo* sequencing task.

244
245
246
247
248
249
250
251



252
253
254

Figure 4: Error injection and reflection training for augmenting reasoning in bio. sequence models.

3.2.1 MODEL ARCHITECTURE

257
258
259
260
261
262
263
264
265
266
267
268
269

We adopt a Transformer-based encoder-decoder architecture, following standard designs for this task (Yilmaz et al., 2023). The input spectrum is denoted as $\mathbf{H} = \{\mathbf{I}, c, m\}$, where $\mathbf{I} = \{(m/z_k, i_k)\}_{k=1}^K$ consists of K mass-to-charge and intensity pairs. Each $(m/z_k, i_k)$ is embedded via a sinusoidal encoding and projected into a d -dimensional vector. The resulting sequence of embeddings forms the encoder output $\mathbf{E} \in \mathbb{R}^{K \times d}$.

The decoder is an autoregressive Transformer with causal self-attention and cross-attention to \mathbf{E} . Let $\mathbf{h}_t^{(i)}$ denote the hidden state at decoding step t and decoder layer i . Causal self-attention ensures that each position only attends to past outputs: $\mathbf{h}_t^{(i)} = \text{Attn}(\mathbf{h}_t^{(i-1)}, \{\mathbf{h}_{1:t-1}^{(i-1)}\})$. Each $\mathbf{h}_t^{(i)}$ then attends to the spectrum features via cross-attention over \mathbf{E} . The final decoder output at step t , denoted $\mathbf{h}_t^{(L)}$, is projected onto the vocabulary $\mathcal{V}_{\text{protein}^+}$: $P_t(\cdot | \mathbf{S}, \mathbf{y}_{<t}) = \text{softmax}(\mathbf{W}\mathbf{h}_t^{(L)})$; Token \mathbf{y}_t is sampled from $P_t(\cdot | \mathbf{S}, \mathbf{y}_{<t})$. Training proceeds via standard next-token prediction (NTP), minimizing the cross-entropy loss over ground-truth sequences.

3.2.2 RANDOM POSITION, RANDOM ERROR (RPRE) INJECTION

For a given input-spectrum and ground-truth sequence pair $(\mathbf{H}, \mathbf{A} = \{a_1, a_2, \dots, a_n\}) \in \mathcal{D}_{\text{de novo}}$, we perform *reflection error injection* (RPRE) to construct training sequences for teaching the model how to reflect on its past generation and correct potential mistakes.

In the RPRE strategy, we randomly sample a position $t \sim \text{Uniform}(1, n)$ and a random token $\tilde{a}_t \sim \mathcal{V}_{\text{protein}}$. We then replace the token at position t with \tilde{a}_t , regardless of correctness, and insert a reflection token $\langle \text{reflect} \rangle$ immediately afterward, followed by the original correct token a_t . The modified training target becomes:

$$\mathbf{A}' = \{a_1, \dots, a_{t-1}, \tilde{a}_t, \langle \text{reflect} \rangle, a_t, a_{t+1}, \dots, a_n\}$$

This approach enables the model to learn how to self-correct through reflection. Notably, when $\tilde{a}_t = a_t$ (i.e., the random token happens to be correct), the reflection becomes a no-op-teaching the model to maintain confidence when no error is present. As a result, RPRE encourages both correction and retention behavior through the same mechanism.

3.2.3 RANDOM POSITION, LOOKAHEAD ERROR (RPLE) INSERTION

While RPRE introduces reflection opportunities, the injected error \tilde{a}_t may be overly implausible (e.g., “1 + 1 = 3⟨reflect⟩2”) and thus trivially detectable. To inject more cognitively challenging errors, we propose *Random Position, Lookahead Error (RPLE)* insertion, which samples replacement tokens within the ground-truth sequence itself.

Given a spectrum-label pair $(\mathbf{H}, \mathbf{A} = \{a_1, a_2, \dots, a_n\}) \in \mathcal{D}_{\text{de novo}}$, we randomly select a position $t \sim \text{Uniform}(1, n)$ and choose an error token \tilde{a}_t from a set of true tokens that appear later in the sequence: $\tilde{a}_t \sim \{a_{t+1}, a_{t+2}, \dots, a_n\}$. We then replace a_t with \tilde{a}_t , append the reflection token $\langle \text{reflect} \rangle$, and restore the correct token a_t . This strategy creates more realistic, sequence-consistent errors and helps the model learn to correct subtle confusions such as token swaps or premature token use—common in peptide prediction (e.g., predicting “BA” instead of “AB”).

3.2.4 BATCH-LEVEL ONLINE DYNAMIC ERROR INJECTION

To prevent the model from memorizing the same sequence and error type and encourage generalizable reflection behavior, we adopt *batch-level online dynamic error injection* during training. For each mini-batch randomly sampled from $\mathcal{D}_{\text{de novo}}$, we apply error injection in real-time to the target sequences.

Given an injection ratio $\alpha \in [0, 1]$, a fraction α of sequences in each batch are modified using either the RPRE or RPLE strategy. The remaining $(1 - \alpha)$ fraction are left unchanged:

Algorithm 1 Online Dynamic Reflection-Error Injection During Training

Require: Batch $\mathcal{B} = \{(\mathbf{H}^{(i)}, \mathbf{A}^{(i)})\}_{i=1}^B$, injection ratio $\alpha \in [0, 1]$

- 1: **for** each (\mathbf{H}, \mathbf{A}) in \mathcal{B} **do**
 - 2: **if** $\text{Uniform}(0, 1) < \alpha$ **then**
 - 3: Sample $t \sim \text{Uniform}(1, |\mathbf{A}|)$, choose RPRE or RPLE
 - 4: $\tilde{a}_t \sim$ **RPRE:** $\mathcal{V}_{\text{protein}}$ **RPLE:** $\{a_{t+1}, \dots, a_n\}$
 - 5: $\mathbf{A}' \leftarrow \{a_1, \dots, a_{t-1}, \tilde{a}_t, \langle \text{reflect} \rangle, a_t, \dots, a_n\}$
 - 6: Replace \mathbf{A} with \mathbf{A}' in \mathcal{B}
 - 7: **end if**
 - 8: **end for**
-

3.2.5 ERROR POSITION GRADIENT BLOCKING

To ensure that injected errors act as *contextual signals* rather than learning targets, we apply **gradient blocking** at error positions during training. Formally, given a modified target sequence $\mathbf{A}' = \{a_1, \dots, \tilde{a}_t, \langle \text{reflect} \rangle, a_t, \dots\}$, we exclude the loss term at position t —corresponding to the injected error token \tilde{a}_t —from the training objective. This allows the model to condition on the error via causal attention and learn to generate the reflection token $\langle \text{reflect} \rangle$ in response, without learning the generation of the incorrect prediction itself (Fig. 4 right panel).

Table 1: Comparison of different settings of reflection pretraining and baseline on 9-Species test set.

Method (beam size = 1)	Mouse	Human	Yeast	<i>M.mazei</i>	Honeybee	Tomato	<i>R.bean</i>	<i>Bacillus</i>	<i>C.bacteria</i>
Standard Pretrain	Amino Acid Precision								
Transformer (Δ baseline)	0.717	0.649	0.752	0.713	0.706	0.763	0.714	0.753	0.66
placeholder Token Baselines									
Transformer + 60% Insertion	0.704	0.633	0.755	0.692	0.680	0.745	0.701	0.726	0.630
Reflect. Finetune w/ RPRE									
Transformer + 60% Error	0.737	0.672	0.736	0.739	0.690	0.784	0.749	0.774	0.684
Reflect. Pretrain w/ RPRE									
Transformer + 60% Error	0.765	0.713	0.796	0.772	0.721	0.806	0.795	0.811	0.713
Reflect. Pretrain w/ RP(RE +LE)									
Transformer + 60% Error	0.784	0.735	0.808	0.786	0.743	0.813	0.808	0.820	0.723
+ 90% Error	0.792	0.752	0.809	0.790	0.744	0.822	0.817	0.826	0.737
	(+10.5%)	(+15.9%)	(+7.6%)	(+10.8%)	(+5.4%)	(+7.7%)	(+14.4%)	(+9.7%)	(+11.2%)
+ 99% Error	0.786	0.739	0.807	0.789	0.742	0.819	0.814	0.823	0.732
Standard Pretrain	Peptide Precision								
Transformer (Δ baseline)	0.443	0.433	0.584	0.522	0.460	0.606	0.652	0.580	0.413
Reflect. Finetune w/ RPRE									
Transformer + 60% Error	0.443	0.434	0.589	0.530	0.465	0.607	0.563	0.583	0.416
Reflect. Pretrain w/ RPRE									
Ours (60% ER, 6 AA/PEP)	0.485	0.494	0.632	0.575	0.507	0.644	0.629	0.635	0.458
Reflect. Pretrain w/ RP(RE +LE)									
Transformer + 60% Error	0.513	0.535	0.651	0.592	0.534	0.658	0.665	0.662	0.470
+ 90% Error	0.533	0.563	0.661	0.605	0.544	0.668	0.657	0.674	0.490
	(+20.3%)	(+30.0%)	(+13.2%)	(+15.9%)	(+18.3%)	(+10.2%)	(+0.8%)	(+16.2%)	(+0.9%)
+ 90% Error	0.515	0.537	0.648	0.595	0.534	0.660	0.657	0.660	0.474

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. Following established protocols (Yilmaz et al., 2023), we use the MassIVE-KB dataset (Wang et al., 2018), which contains **30 million peptide-spectrum matches (PSMs)** collected from diverse mass spectrometry platforms. Evaluation is conducted on the widely adopted 9-species-v1 and 9-species-v2 benchmarks.

Implementation. Input peaks and amino acids are embedded into 512-dimensional vectors. Both the spectrum encoder and peptide decoder are 9-layer Transformers with 8 attention heads and 1024-dimensional hidden states. Models are trained on eight NVIDIA A100 80GB GPUs using AdamW ($\alpha_0 = 5 \times 10^{-4}$) with a 10k-step linear warmup followed by cosine decay.

Evaluation Metrics. We report two metrics. **Amino acid-level (AA) Precision** considers a prediction correct if the amino acid token aligns with the target position. Accuracy is defined as M_{AA}/T_{AA} , where M_{AA} is the number of matched amino acids and T_{AA} is the total number of predicted amino acids. **Peptide-level precision** considers a prediction correct only if the entire predicted peptide matches the ground truth sequence exactly. Precision is defined as M_{pep}/T_{pep} , where M_{pep} is the number of matched peptides and T_{pep} is the total number of predicted peptides.

4.2 RESULTS

Results Analysis. As shown in Table 1, reflection-based pretraining leads to substantial improvements in both amino acid and peptide precision across all 9 species. In contrast, **finetuning** shows negligible gains over baselines, and reflection tokens are never used during inference (Table 3), indicating that reflection behavior is not learned through finetuning alone.

Notably, increasing the error ratio from 60% to 90% further boosts peptide precision in all species, while also increasing the frequency of reflection token usage from 2.3% to 4.46% during inference (Table 3). Despite most `<reflect>` tokens retaining the original answer (e.g., 67.8% in the 90% error model), this indicates a learned capacity for self-assessment. Moreover, incorporating RPLE during pretraining introduces more realistic, confusing errors, pushing peptide precision further up.

Table 2: Comparison among bio-inspired methods and reflection-reflection-pretrained model.

Method	Mouse	Human	Yeast	M.mazei	Honeybee	Tomato	R.bean	Bacillus	C.bacteria	Average
AA Precision										
Bio. Inspired Methods										
Peaks	0.600	0.639	0.748	0.673	0.633	0.728	0.644	0.719	0.586	0.663
DeepNovo	0.623	0.610	0.750	0.694	0.630	0.731	0.679	0.742	0.602	0.673
PointNovo	0.626	0.606	0.779	0.712	0.644	0.733	0.730	0.768	0.589	0.688
InstaNovo	0.703	0.636	0.691	0.712	0.660	0.732	0.711	0.739	0.619	0.689
Casanovo, Beam = 1	0.717	0.649	0.752	0.713	0.706	0.763	0.714	0.753	0.663	0.704
HelixNovo, Beam = 1	0.750	0.648	0.758	0.766	0.699	0.757	0.771	0.799	0.661	0.734
Reflection Pretrain										
90% Error, Beam = 1	0.792	0.752	0.809	0.790	0.744	0.822	0.817	0.826	0.737	0.788
90% Error, Beam = 3	0.799	0.763	0.818	0.800	0.755	0.827	0.832	0.836	0.745	0.797
90% Error, Beam = 5	0.805	0.774	0.826	0.810	0.766	0.831	0.846	0.846	0.753	0.806
Peptide Recall										
Bio. Inspired Methods										
Peaks	0.197	0.277	0.428	0.356	0.287	0.403	0.362	0.387	0.203	0.333
DeepNovo	0.286	0.293	0.462	0.422	0.330	0.454	0.436	0.449	0.253	0.387
PointNovo	0.355	0.351	0.534	0.478	0.396	0.513	0.511	0.518	0.298	0.439
InstaNovo	0.471	0.455	0.559	0.528	0.466	0.732	0.564	0.576	0.416	0.530
Casanovo	0.443	0.433	0.584	0.522	0.460	0.606	0.652	0.580	0.413	0.521
Reflection Pretrain										
90% Error, Beam=1	0.533	0.563	0.661	0.605	0.544	0.668	0.657	0.674	0.490	0.600
90% Error, Beam=3	0.540	0.573	0.669	0.615	0.555	0.674	0.667	0.685	0.497	0.608
90% Error, Beam=5	0.546	0.582	0.676	0.624	0.565	0.680	0.676	0.696	0.504	0.617

Case Study: Self-Reflection and Error-Correction During Inference Time

Raw prediction: R<reflect>KYFHNELM+15.995NYVQEC+57.021QFDSETSL\$
Post-processed output: KYFHNELM+15.995NYVQEC+57.021QFDSETSL
Ground-truth label: KYFHNELM+15.995NYVQEC+57.021QFDSETSL

Case Study: Self-Reflection but no-op During Inference time

Raw prediction: KDFFTYME<reflect>E\$
Post-processed output: KDFFTYME
Ground-truth label: KDFFTYME

Outperforming Bio-Inspired Models Without Domain Modules. As shown in Table ??, our reflection-pretrained model achieves state of the art in peptide sequencing task, **surpassing all bio-inspired baselines**, despite using **no task-specific architectural modules**. Built on a standard Transformer, it leverages the expressive power of reflection-driven Chain-of-Thought reasoning, demonstrating that procedural reasoning can outperform hard-coded biological priors.

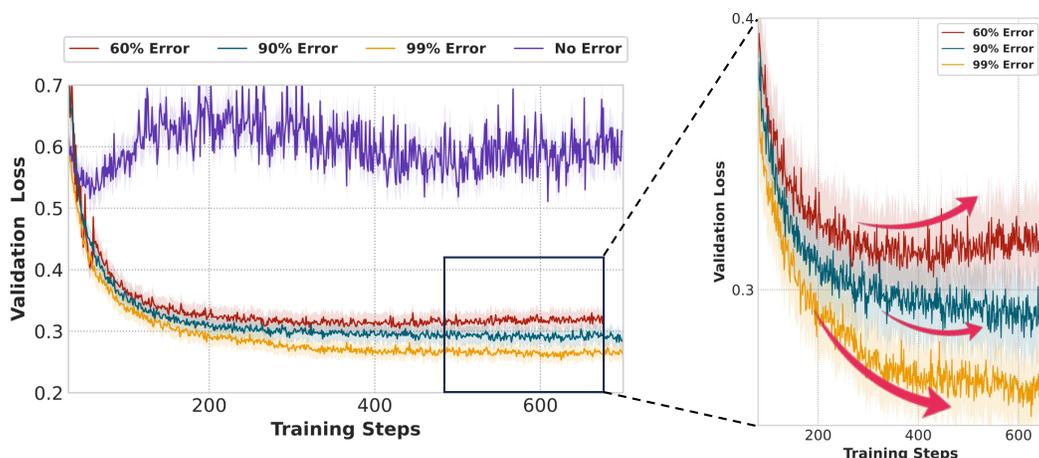
Reflection Behavior Case Study. As shown in Table of Case studies, the model exhibits **strong reflection** capabilities during inference. It successfully detects and corrects its own mistakes using <reflect>, and equally important, it chooses to retain correct outputs when no revision is needed. This balance between self-correction and confident continuation highlights the model’s ability to exercise self-doubt as well as self-affirmation.

Impact of Error Injection on Generalization. Figure 5 shows the effect of different reflection error ratios on validation loss during pretraining. Without error injection (blue curve), the model exhibits clear signs of overfitting: validation loss initially decreases but begins to rise after 100 steps. In contrast, injecting reflection errors—especially at 90% and 99% rates—leads to more stable and monotonic decreases in validation loss. This improvement stems from the fact that our **batch-dynamic error** alters sequences throughout the training, preventing the model from memo-

Table 3: Reflection usage during inference (Among incorrect sequences). **Use:** % sequence uses reflection token among incorrect sequences if reflection were banned; **Corr.:** % reflections that correct the errors among all usage in incorrect predictions; **Same:** % reflections that retained the original error among all incorrect predictions. **The rest are changing one error to another error.**

Model	Use	Corr.	Same
Baseline	0.0%	–	–
Reflect FT	0.0%	0.0%	0.0%
PT 60% Error	10.9%	16.0%	69.9%
PT 90% Error	16.8%	14.1%	65.2%

432 rizing fixed input-output mappings. As a result, the model is forced to generalize across perturbed
 433 inputs, reducing overfitting.
 434



441 Figure 5: Error Injection Rate on Validation Loss. Loss Plotting and Logging by Neptune.ai.
 442

453 **Human-in-the-Loop Reflection.** We manually picked 10 sequences where the model failed to
 454 detect errors. By manually inserting `<reflect>` tokens at the error positions and then generating
 455 from there, we found that 2 out of 10 sequences were successfully corrected during inference. While
 456 this approach is not yet scalable—especially in real-world settings lacking ground-truth labels—it
 457 highlights a promising opportunity for expert biologists to intervene and guide reflective reasoning.
 458

459 Additional results, including evaluation on extended benchmark datasets, the effect of beam size, and
 460 further case studies with analysis, are provided in Appendix.

461 Case Study: Manual Reflection Improves Prediction

463 Raw prediction:	RLANLYWLS
464 Manual intervention:	RL<reflect>
465 Output after intervention:	RL<reflect>MNFYGFL
466 Ground-truth label:	RMNFYGFL

468 **Human-Guided Reflection for Correction.** Case study in Table 4.2 demonstrates the potential of
 469 human-in-the-loop intervention using reflection tokens. As shown above, the model initially produces
 470 an incorrect prediction (“RLANLYWL”), but upon manually inserting a `<reflect>` token after the
 471 first residue, the model self-corrects and generates the correct target sequence (“RMNFYGFL”).
 472

473 Notably, the correction cascades beyond a single token—the model adjusts not just the second residue
 474 but the entire downstream sequence. This highlights the model’s capacity for global reasoning
 475 triggered by a local reflective signal. Although we do not yet have automated methods for locating
 476 such errors in unlabeled real-world data, this opens the door for expert-guided inference: domain
 477 experts can flag uncertain regions and inject `<reflect>` tokens to improve model output. This
 478 enables a promising direction for collaborative, controllable biological sequence generation.

480 5 CONCLUSION

481
 482 We introduce a reflection-based pretraining approach that equips biological sequence models with
 483 intermediate reasoning abilities, inspired by Chain-of-Thought prompting. This improves expressive-
 484 ness, encourages self-correction, boosts accuracy, and reduces overfitting—particularly for *de novo*
 485 peptide sequencing. The framework also enables human-in-the-loop use, bridging biological and
 natural language models and highlighting the potential of reasoning-driven modeling in biology.

6 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our research, all experiments are conducted on the publicly available MassIVE-KB dataset (Wang et al., 2018), with evaluations performed on the standard 9-species benchmarks. Our proposed reflection pretraining framework, including the RPRE and RPLE error injection strategies, dynamic data updating, and gradient blocking, is described in detail in Section 3 with Code available in the provided link. The model is a standard Transformer encoder-decoder, and all architectural details and training hyperparameters—such as learning rates, optimizer, and hardware configuration—are specified in Section 4.1 and the Appendix. To further facilitate replication, we released our source code and model weights in the abstract section with a anonymous link, with sample reflection output also provided there. Together, these resources provide a comprehensive basis for verifying our results and extending our work.

REFERENCES

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928): 198–207, 2003a. ISSN 1476-4687. URL <https://doi.org/10.1038/nature01511>.
- Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928): 198–207, 2003b.
- Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.
- Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Denis Beslic, Georg Tscheuschner, Bernhard Y Renard, Michael G Weller, and Thilo Muth. Comprehensive evaluation of peptide de novo sequencing tools for monoclonal antibody assembly. *Briefings in Bioinformatics*, 5:1–12, dec 2022. ISSN 1467-5463.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.
- Jürgen Cox and Matthias Mann. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26(12):1367–1372, 2008. ISSN 10870156.
- Vlado Dančák, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology*, 6(3-4): 327–342, 1999.
- Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.
- Zhi-Luo Deng, Philipp C Münch, René Mreches, and Alice C McHardy. Rapid and accurate identification of ribosomal rna sequences via deep learning. *Nucleic acids research*, 50(10): e60–e60, 2022.
- Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*, 2022.

- 540 Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehaw, Yu Wang, Llion Jones, Tom
541 Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding
542 the language of life through self-supervised learning. *IEEE transactions on pattern analysis and*
543 *machine intelligence*, 44(10):7112–7127, 2021.
- 544 Kevin Eloff, Konstantinos Kalogeropoulos, Oliver Morell, Amandla Mabona, Jakob Berg Jespersen,
545 Wesley Williams, Sam PB van Beljouw, Marcin Skwark, Andreas Hougaard Laustsen, Stan JJ
546 Brouns, et al. De novo peptide sequencing with instanovo: Accurate, database-free peptide
547 identification for large scale proteomics experiments. *bioRxiv*, pp. 2023–08, 2023.
- 549 J K Eng, A L McCormack, and J R Yates. An approach to correlate tandem mass spectral data of
550 peptides with amino acid sequences in a protein database. *Journal of the American Society for*
551 *Mass Spectrometry*, 5(11):976–989, nov 1994. ISSN 1044-0305 (Print).
- 552 Ari Frank and Pavel Pevzner. Pepnovo: de novo peptide sequencing via probabilistic network
553 modeling. *Analytical chemistry*, 77(4):964–973, 2005.
- 555 Erik Garrison. Memory makes computation universal, remember? *arXiv preprint arXiv:2412.17794*,
556 2024.
- 557 Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive
558 neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.
- 560 Robert L Hettich, Chongle Pan, Karuna Chourey, and Richard J Giannone. Metaproteomics: harness-
561 ing the power of high performance mass spectrometry to identify the suite of proteins that control
562 metabolic activities in microbial communities. *Analytical chemistry*, 85(9):4203–4214, may 2013.
563 ISSN 1520-6882 (Electronic).
- 564 Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing
565 prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*,
566 2023.
- 568 Zhi Jin, Sheng Xu, Xiang Zhang, Tianze Ling, Nanqing Dong, Wanli Ouyang, Zhiqiang Gao, Cheng
569 Chang, and Siqi Sun. Contranovo: A contrastive learning approach to enhance de novo peptide
570 sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp.
571 144–152, 2024.
- 572 John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger,
573 Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate
574 protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.
- 576 Korrawe Karunratanakul, Hsin-Yao Tang, David W Speicher, Ekapol Chuangsuwanich, and Sira
577 Sriswasdi. Uncovering Thousands of New Peptides with Sequence-Mask-Search Hybrid De Novo
578 Peptide Sequencing Framework. *Molecular & cellular proteomics : MCP*, 18(12):2478–2491, dec
579 2019. ISSN 1535-9484 (Electronic).
- 580 Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish
581 Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv*
582 *preprint arXiv:2210.02406*, 2022.
- 584 Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*
585 *arXiv:1412.6980*, 2014.
- 586 Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large
587 language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:
588 22199–22213, 2022.
- 590 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444,
591 2015.
- 592 Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to
593 solve inherently serial problems. *arXiv preprint arXiv:2402.12875*, 1, 2024.

- 594 Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin,
595 Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level
596 protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
597
- 598 Jieyi Long. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*, 2023.
599
- 600 Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiayi Cui, Calvin Yu-Chian Chen, Li Yuan,
601 and Yonghong Tian. Prollama: A protein large language model for multi-task protein language
602 processing. *IEEE Transactions on Artificial Intelligence*, 2025.
- 603 Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby,
604 and Gilles Lajoie. Peaks: powerful software for peptide de novo sequencing by tandem mass
605 spectrometry. *Rapid communications in mass spectrometry*, 17(20):2337–2342, 2003.
606
- 607 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri
608 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement
609 with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- 610 William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought.
611 *arXiv preprint arXiv:2310.07923*, 2023.
612
- 613 Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman,
614 Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language
615 models. *arXiv preprint arXiv:2307.06435*, 2023.
- 616 Cheuk Chi A Ng, Yin Zhou, and Zhong-Ping Yao. Algorithms for de-novo sequencing of peptides by
617 tandem mass spectrometry: A review. *Analytica Chimica Acta*, pp. 341330, 2023.
618
- 619 Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David
620 Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work:
621 Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*,
622 2021.
- 623 Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine
624 Learning Research*, 22(75):1–35, 2021.
625
- 626 D N Perkins, D J Pappin, D M Creasy, and J S Cottrell. Probability-based protein identification by
627 searching sequence databases using mass spectrometry data. *Electrophoresis*, 20, 1999. URL
628 <http://dx.doi.org/3.0.CO;2-2>.
- 629 Rui Qiao, Ngoc Hieu Tran, Lei Xin, Xin Chen, Ming Li, Baozhen Shan, and Ali Ghodsi. Computa-
630 tionally instrument-resolution-independent de novo peptide sequencing for high-resolution devices.
631 *Nature Machine Intelligence*, 3(5):420–425, 2021.
632
- 633 Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu,
634 and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pp.
635 8844–8856. PMLR, 2021.
- 636 Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo,
637 Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from
638 scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National
639 Academy of Sciences*, 118(15):e2016239118, 2021.
640
- 641 Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of thoughts:
642 Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*, 2023.
- 643 Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages
644 can transformers express? a survey. *Transactions of the Association for Computational Linguistics*,
645 12:543–561, 2024.
646
- 647 Mirac Suzgun and Adam Tauman Kalai. Meta-prompting: Enhancing language models with task-
agnostic scaffolding. *arXiv preprint arXiv:2401.12954*, 2024.

- 648 Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing
649 by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.
- 650
- 651 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz
652 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing*
653 *systems*, 30, 2017.
- 654
- 655 Mingxun Wang, Jian Wang, Jeremy Carver, Benjamin S Pullman, Seong Won Cha, and Nuno
656 Bandeira. Assembling the community-scale discoverable human proteome. *Cell systems*, 7(4):
657 412–421, 2018.
- 658
- 659 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
660 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in*
661 *neural information processing systems*, 35:24824–24837, 2022.
- 662
- 663 Jun Xia, Shaorong Chen, Jingbo Zhou, Tianze Ling, Wenjie Du, Sizhe Liu, and Stan Z Li. Adanovo:
664 Adaptive\emph {De Novo} peptide sequencing with conditional mutual information. *arXiv*
665 *preprint arXiv:2403.07013*, 2024.
- 666
- 667 Yijia Xiao, Wanxia Zhao, Junkai Zhang, Yiqiao Jin, Han Zhang, Zhicheng Ren, Renliang Sun, Haixin
668 Wang, Guancheng Wan, Pan Lu, et al. Protein large language models: A comprehensive survey.
669 *arXiv preprint arXiv:2502.17504*, 2025.
- 670
- 671 Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. A survey
672 on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions*
673 *on Pattern Analysis and Machine Intelligence*, 45(10):11407–11427, 2023.
- 674
- 675 Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and
676 Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of
677 single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.
- 678
- 679 Tingpeng Yang, Tianze Ling, Boyan Sun, Zhendong Liang, Fan Xu, Xiansong Huang, Linhai Xie,
680 Yonghong He, Leyuan Li, Fuchu He, et al. Introducing π -helixnovo for practical large-scale de
681 novo peptide sequencing. *Briefings in Bioinformatics*, 25(2):bbae021, 2024.
- 682
- 683 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan.
684 Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural*
685 *Information Processing Systems*, 36, 2024.
- 686
- 687 Melih Yilmaz, William Fondrie, Wout Bittremieux, Sewoong Oh, and William S Noble. De novo
688 mass spectrometry peptide sequencing with a transformer model. In *International Conference on*
689 *Machine Learning*, pp. 25514–25522. PMLR, 2022.
- 690
- 691 Melih Yilmaz, William E Fondrie, Wout Bittremieux, Carlo F Melendez, Rowan Nelson, Varun
692 Ananth, Sewoong Oh, and William Stafford Noble. Sequence-to-sequence translation from mass
693 spectra to peptides with a transformer model. *BioRxiv*, pp. 2023–01, 2023.
- 694
- 695 Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with
696 reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.
- 697
- 698 Jing Zhang, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng
699 Zhang, Gilles A Lajoie, and Bin Ma. PEAKS DB: de novo sequencing assisted database search
700 for sensitive and accurate peptide identification. *Molecular & cellular proteomics : MCP*, 11(4):
701 M111.010587, apr 2012. ISSN 1535-9484 (Electronic).

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

APPENDIX

A RELATED WORK

Chain-of-Thought Reasoning in Natural Language Models. Traditional natural language processing (NLP) systems predominantly followed an *answer-only* generation paradigm, directly mapping inputs to final outputs. This changed with the introduction of Chain-of-Thought (CoT) prompting (Wei et al., 2022), which enables models to emit intermediate reasoning steps-non-answer tokens-prior to producing a final answer. CoT significantly improves performance on tasks requiring compositional reasoning (Nye et al., 2021; Kojima et al., 2022), by allowing models to retain transient memory, perform iterative updates, and correct earlier errors during generation. The success of CoT has led to a surge in structured reasoning approaches, including tree-structured (Yao et al., 2024; Long, 2023), graph-based (Besta et al., 2024; Sel et al., 2023), and decompositional methods (Zhou et al., 2022; Drozdov et al., 2022; Khot et al., 2022). Further extensions (Zelikman et al., 2022; Madaan et al., 2024; Suzgun & Kalai, 2024) demonstrate that the effectiveness of these methods hinges on the high expressiveness of human language-capable of articulating logic, uncertainty, abstraction, and reflection of which enrich the non-answer space.

Limitations of Current Biological Sequence Generation. Deep learning models have advanced sequence modeling in biology across DNA (Alipanahi et al., 2015; Zhang et al., 2021), RNA (Yang et al., 2022; Deng et al., 2022), and proteins (Rives et al., 2021; Jumper et al., 2021; Elnaggar et al., 2021; Lin et al., 2023). Yet, despite these advances, generation in biological domains-particularly in *de novo* peptide sequencing (Yilmaz et al., 2022; Eloff et al., 2023; Xia et al., 2024)-remains constrained to answer-only prediction. These models operate over token spaces (e.g., 20 amino acids) with inherently limited semantic richness, producing final sequence tokens without externalized reasoning traces. We argue that this limitation is not merely architectural but stems from the low expressive capacity of biological sequence languages. Unlike natural language, biological alphabets are poorly suited for encoding intermediate states, error attribution, or hypothetical exploration capabilities central to CoT-style reasoning.

Expressiveness, Computational Power, and Generative Capacity. While Transformers with CoT prompting can theoretically simulate Turing-complete machines (Pérez et al., 2021; Merrill & Sabharwal, 2023; Strobl et al., 2024), their practical reasoning capacity is constrained by input length, memory bandwidth (Arora & Barak, 2009; Garrison, 2024), and crucially, the expressiveness of the output token space. In biological models, the inability to produce diverse non-answer tokens impedes the model’s ability to represent internal state transitions or conduct hypothesis-driven exploration. This constitutes a fundamental bottleneck, distinct from issues of model size or architecture. In contrast, natural language models such as GPT (Achiam et al., 2023) benefit from an output space rich in semantics, enabling a flexible reasoning substrate. Biological models like ESM (Rives et al., 2021) or ProLLaMA (Lv et al., 2025) lack this capacity, as their generation is grounded in symbolic alphabets that are not optimized for reasoning abstraction.

Embedding Reflection and Self-Correction into Biological Models. Recent advances in NLP promote self-reflection and critique during inference (Madaan et al., 2024), but these methods are predominantly post hoc and not integrated into model training. Our work is the first to incorporate reflection and error correction directly into the pretraining phase of biological sequence models.

Specifically, we propose a reflection-based pretraining strategy for *de novo* peptide sequencing, where models are exposed to corrupted prediction traces and explicitly trained to identify and amend errors. This goes beyond inference-time prompting, embedding reasoning patterns into model weights. Complementary efforts in scaling Transformer context (Beltagy et al., 2020; Zhang et al., 2023; Jiang et al., 2023) address sequence length limitations but do not address the deeper issue of representational constraints. By introducing a structured set of non-answer tokens and designing tasks that invoke reasoning within biological domains, our framework redefines the generative capacity of biological models to support more intelligent, interpretable, and compositional generation.

B MODEL EXPRESSIVE POWER: FROM MLPs TO RECURRENT ARCHITECTURES AND TRANSFORMERS

The **expressive power** of a neural network architecture dictates its fundamental capacity to represent or approximate complex functions, thereby determining its suitability for a diverse range of computational tasks. This capacity is intrinsically linked to the architectural design, especially its mechanisms for information propagation and transformation across processing stages. Architectures with constrained expressive power may falter on tasks demanding deep sequential reasoning or intricate dependency modeling, whereas more expressive models can capture a richer set of complex patterns and computational processes.

B.1 DEFINING EFFECTIVE COMPUTATIONAL DEPTH

We propose to formalize a critical dimension of expressive power via the concept of *Effective Computational Depth*, denoted $D_{\text{eff}}(\mathcal{M})$. This metric quantifies the maximal length L of a sequence of non-trivial transformations $(\mathcal{T}_i)_{i=1}^L$ that a model \mathcal{M} (with parameters θ) can compose to map an input x to an output $y^*(x)$, potentially averaged over a distribution of tasks $\mathcal{P}_{\text{task}}$. Each \mathcal{T}_i represents a distinct computational step, possibly conditioned on intermediate results generated by preceding transformations $\mathcal{T}_{j < i}$. Formally:

$$D_{\text{eff}}(\mathcal{M}) := \sup \left\{ L \in \mathbb{N} \mid \exists \mathbf{T}^{(L)} \subseteq \mathcal{M} \text{ s.t. } \left(\forall P \in \mathcal{P}_{\text{task}}^{(L)}, \mathcal{R}_{\min}(P, \mathbf{T}^{(L)}) \leq \epsilon \right) \right\} \quad (1)$$

Here, $\mathcal{P}_{\text{task}}^{(L)}$ signifies tasks inherently requiring L sequential computational steps. \mathcal{R}_{\min} represents the minimum achievable risk (or loss) for task P using the sequence of L transformations $\mathbf{T}^{(L)}$ afforded by model \mathcal{M} , and ϵ is a small tolerance indicating successful task resolution. A higher $D_{\text{eff}}(\mathcal{M})$ signifies a greater capacity for deep sequential processing, a cornerstone of tackling complex computations. The depth complexity, as discussed in prior work, measures the number of sequential steps after considering all parallel processing a model performs. $D_{\text{eff}}(\mathcal{M})$ aims to capture this notion of maximal sequential transformative capacity.

B.2 MULTI-LAYER PERCEPTRONS (MLPs)

Multi-Layer Perceptrons (MLPs) are structured with a fixed number of layers, say m . Computation proceeds sequentially through these layers: $h^{(i)} = \sigma(W^{(i)}h^{(i-1)})$. Each layer’s operation constitutes a transformation \mathcal{T}_i . Since m is a constant, independent of the input sequence length n , the effective computational depth $D_{\text{eff}}(\text{MLP})$ is $O(m)$. This effectively becomes $O(1)$ with respect to n , as m does not scale with input size. This inherently fixed and often shallow depth restricts MLPs from adeptly solving tasks that necessitate iterative refinement or the processing of sequences with variable and potentially long-range dependencies.

B.3 RECURRENT NEURAL NETWORKS (RNNs)

Recurrent Neural Networks (RNNs) incorporate recurrent connections, enabling the output from a previous time step h_{t-1} to be an input to the current step: $h_t = g_{\theta}(h_{t-1}, x_t)$. This architectural feature allows information to persist and be transformed across a number of time steps that can scale with the input sequence length n . In this context, each transformation \mathcal{T}_i corresponds to the recurrent computation at a given time step. Consequently, the effective computational depth $D_{\text{eff}}(\text{RNN})$ can reach $O(n)$. This substantial increase in sequential depth capacity empowers RNNs to effectively

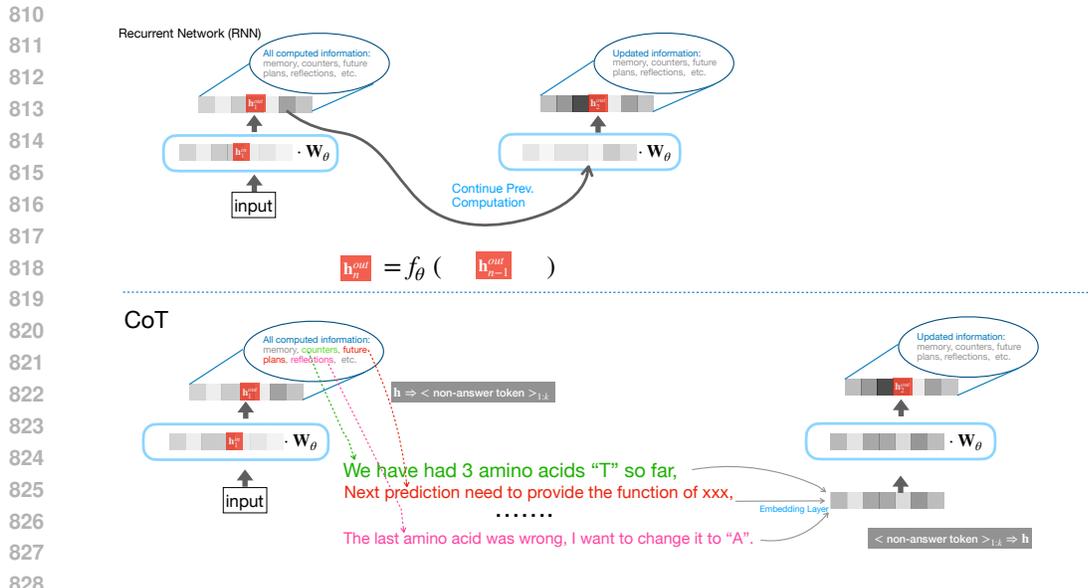


Figure 6: Illustration of how CoT approximate recurrent computation in LMs.

model sequential data and tasks characterized by temporal dependencies, such as string manipulations or sequential decision-making, far surpassing the capabilities of standard MLPs. RNNs are considered “recurrence-complete” as they possess the capability to simulate any one-term recurrent function $h_t = g'(h_{t-1})$, given adequate network capacity and appropriate non-linear activation functions, a property underpinned by the Universal Approximation Theorem.

B.4 TRANSFORMERS

Standard Transformer architectures, despite their considerable success driven by the attention mechanism, operate with a fixed number of layers, m , akin to MLPs. While attention allows for parallel processing of tokens within each layer and sophisticated information integration from the entire input sequence $x_{1:t}$, the number of sequential transformation steps (i.e., layers) remains constant. The final layer output $h_t^{(m)}$ is a function of the input $x_{1:t}$ but not directly of a temporally prior hidden state $h_{t-1}^{(m)}$ in the same way as in an RNN. The transformations \mathcal{T}_i in a Transformer correspond to the operations within each of its m layers. Therefore, the $D_{\text{eff}}(\text{Transformer})$ is $O(m)$, which is $O(1)$ with respect to the input sequence length n . This characteristic limits their inherent ability to solve tasks that require a computational depth greater than m without architectural modifications or prompting strategies that simulate deeper recurrence.

C CHAIN OF THOUGHT: TOWARDS UNBOUNDED EXPRESSIVE POWER IN THEORY

The advent of Chain of Thought (CoT) prompting has been empirically shown to significantly enhance the reasoning capabilities of Large Language Models (LLMs). Beyond a mere prompting strategy, CoT can be interpreted from a computational theory perspective as a mechanism that fundamentally reshapes the Transformer’s operational dynamics at runtime. In theory, CoT endows autoregressive LLMs with the potential for unbounded expressive power by dynamically altering their computational depth, time complexity, and effective memory.

C.1 AUGMENTING COMPUTATIONAL DEPTH VIA SIMULATED RECURRENCE

Standard Transformer architectures possess a fixed number of layers, m , constraining their *Effective Computational Depth*, $D_{\text{eff}}(\text{Transformer})$, to $O(m)$, or $O(1)$ with respect to input sequence length

864 n . This architectural limitation inherently restricts their capacity for tasks demanding extensive
865 sequential reasoning steps exceeding m .

866 Chain of Thought circumvents this limitation by simulating a recurrent computational process. The
867 core mechanism involves:

- 869 1. Encoding the model’s internal computational state (hidden state h) at the end of a reasoning
870 step into a sequence of natural language tokens (o_1, o_2, \dots, o_k) that form the “thought”. This
871 can be represented as $h \rightarrow o_{1:k}$.
- 872 2. Appending these tokens to the ongoing input sequence, effectively creating an extended
873 context $x_{n+k} = (x_1, \dots, x_n, o_1, \dots, o_k)$.
- 874 3. The model then processes this extended sequence, and the CoT string $o_{1:k}$ is decoded back
875 into a computational state h' , effectively $o_{1:k} \rightarrow h'$.

877 This iterative process, $h^{(t)} \rightarrow o_{1:k}^{(t)} \rightarrow h^{(t+1)}$, emulates the recurrent connection $h_t = g(h_{t-1}, x_t)$
878 found in RNNs, where the model can resume and refine its computation based on the explicitly
879 articulated intermediate state. This transformation allows the internal hidden state h , containing
880 comprehensive computational information, to be first externalized into a sequence of “non-answer
881 tokens” (the CoT string). These tokens, which can articulate ongoing reasoning, memory states,
882 or even self-corrections, are then embedded and re-processed by the model to inform subsequent
883 computations, thereby achieving the $o_{1:k} \rightarrow h'$ step. Consequently, the effective computational depth
884 is no longer rigidly tied to the number of physical layers but rather to the length of the generated
885 thought sequence, $T(n)$. The $D_{\text{eff}}(\text{LLM}+\text{CoT})$ can thus become $O(T(n))$, theoretically allowing
886 for arbitrarily deep sequential computation, limited primarily by factors like context window length
887 or coherence of the generated thought, rather than by fixed architectural depth. This approximated
888 recurrence is what notably improves the model’s computational capacity.

889 C.2 IMPACT ON COMPUTATIONAL TIME

891 The simulation of recurrence through CoT inherently impacts computational time. Standard autore-
892 gressive generation processes one token at a time. By extending the input sequence with intermediate
893 thought tokens $o_{1:k}$, the CoT process necessitates additional computational steps for both generating
894 these thought tokens and subsequently processing them as part of the augmented input. If an input of
895 length n requires $T(n)$ CoT steps (where each step might involve generating multiple tokens), the
896 overall time complexity for processing is increased. The time complexity is enhanced to approxi-
897 mately $O(n + T(n)_{\text{tokens}})$, where $T(n)_{\text{tokens}}$ represents the total number of tokens generated as part
898 of the chain of thought. While this increases the computational cost, it is this very extension that
899 facilitates the enhanced reasoning depth.

900 C.3 EXPANDING EFFECTIVE MEMORY THROUGH ARTICULATED STATES

902 Transformers, through their attention mechanism, can access all previous tokens in their context
903 window. However, the critical information from intermediate computational steps might be diffusely
904 encoded within hidden states and not directly accessible or interpretable for subsequent, distinct
905 reasoning phases that exceed the fixed-layer depth.

906 CoT provides a mechanism to externalize these intermediate computational states. By articulating the
907 hidden state h (or critical aspects of it, such as memory, counters, or reflections) into natural language
908 strings $o_{1:k}$, these strings act as an explicit, persistent memory trace within the context window. The
909 model can then attend to these textual representations of its prior states in subsequent steps. This
910 allows for a form of “read-write” memory capability where the model “writes” its state as text and
911 “reads” it back to continue computation. This capability is crucial for tasks that require maintaining
912 and manipulating information over extended reasoning chains, effectively emulating more powerful
913 memory structures. For instance, LLMs with CoT can achieve tape-like memory access via attention
914 on the generated text, enabling them to tackle tasks at higher levels of the Chomsky hierarchy, such
915 as context-sensitive tasks, which are typically beyond the reach of standard Transformers without
916 CoT. The expressiveness of natural language is posited to be sufficiently universal to encode diverse
917 types of information, including reasoning states, memory, and intermediate results, which is a critical
assumption for this mechanism’s success.

In essence, CoT theoretically empowers Transformers by allowing them to dynamically construct deeper computational graphs and manage more explicit memory traces at runtime, thereby transcending some of their inherent architectural limitations.

D WHY AND WHAT’S *De Novo* PEPTIDE SEQUENCING?

Peptide sequencing from tandem mass spectrometry (MS/MS) data is a cornerstone of proteomics (Aebersold & Mann, 2003a), driving advancements from fundamental biological discovery to drug development (Aebersold & Mann, 2003b; Ng et al., 2023). The goal is to determine the amino acid sequence of peptides directly from their mass spectra. Traditional database search algorithms (Eng et al., 1994; Perkins et al., 1999; Cox & Mann, 2008; Zhang et al., 2012), though widely used, cannot identify novel peptides absent from reference databases. This limitation impedes progress in emerging applications such as *de novo* antibody characterization (Beslic et al., 2022), neoantigen discovery (Karunratanakul et al., 2019), and metaproteomics (Hettich et al., 2013). In contrast, *de novo* peptide sequencing infers sequences without relying on any database, making it indispensable for comprehensive and unbiased proteomic analysis.

The advent of deep learning (LeCun et al., 2015) has transformed *de novo* sequencing, surpassing traditional algorithmic approaches (Dančik et al., 1999; Ma et al., 2003; Frank & Pevzner, 2005). Early models such as DeepNovo (Tran et al., 2017) employed CNNs and LSTMs, while more recent efforts have shifted to Transformer-based architectures (Vaswani et al., 2017), which offer improved scalability and performance. These models generally fall into two categories: autoregressive (AT) and non-autoregressive (NAT). AT models—such as Casanovo (Yilmaz et al., 2022; 2023) and its derivatives (e.g., AdaNovo (Xia et al., 2024), HelixNovo (Yang et al., 2024), InstaNovo (Eloff et al., 2023), ContraNovo (Jin et al., 2024))—predict one amino acid at a time, often achieving high precision through architectural refinements. NAT models like PrimeNovo (Zhang et al., 2024b), on the other hand, generate sequences in parallel, enabling bidirectional context and faster inference (Gu et al., 2017; Xiao et al., 2023). Despite this progress, both paradigms largely lack the capacity for explicit intermediate reasoning, which may limit generalization in noisy or uncertain scenarios.

In this work, we propose *Reflection-Pretraining*, a novel pretraining strategy that enables models to generate intermediate reasoning steps, bringing CoT-style capabilities to domains beyond natural language. Rather than predicting the final output directly, *Reflection-Pretraining* encourages the model to emit structured intermediate tokens that guide and justify the final prediction. By applying this method to the *de novo* sequencing setting, we demonstrate that structured reasoning improves generalization, robustness, and interpretability across peptide prediction tasks.

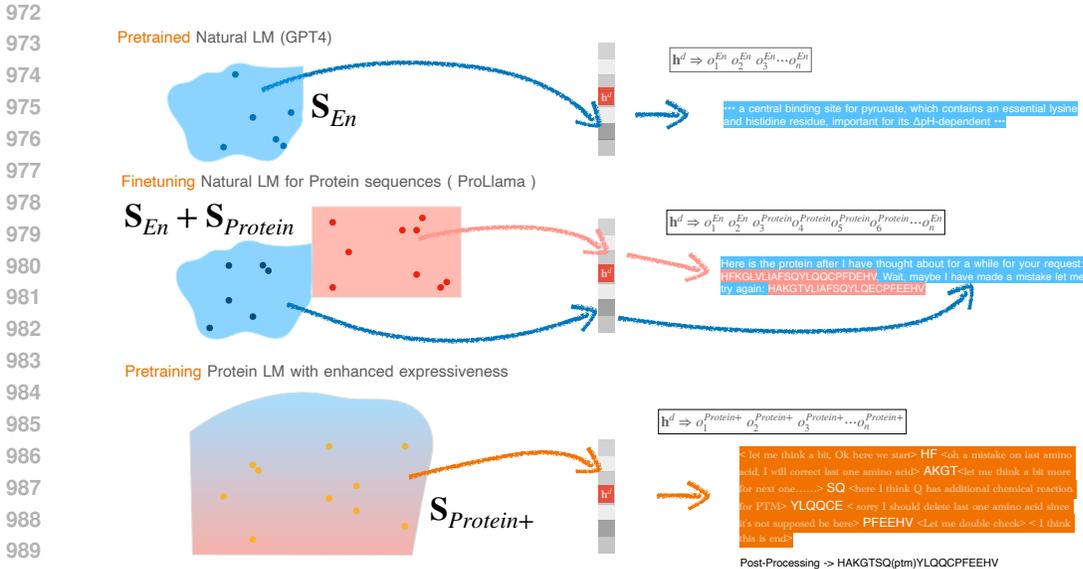
We identify ***de novo* peptide sequencing** as an ideal task for exploring structured neural reasoning for five key reasons:

- It is a foundational problem in proteomics, as peptide sequencing remains the primary method for identifying naturally occurring amino acid sequences.
- The task is inherently reasoning-intensive, requiring interpretation of complex spectral patterns based on biochemical rules, well aligned with the Chain-of-Thought (CoT) paradigm.
- Large-scale, high-quality training datasets of spectra paired with peptide sequences are readily available.
- Evaluation is exact and unambiguous, with each spectrum matched to a ground-truth sequence, unlike more subjective tasks such as language modeling or structure prediction.
- Each predicted token can represent a distinct reasoning step in the sequence inference process, enabling fine-grained supervision and reflection.

These properties make the task uniquely suited to evaluate models that go beyond final-answer generation and instead perform stepwise reasoning under uncertainty.

In the *de novo* peptide sequencing task, the model is given a spectrum instance $\mathbf{H} = \{\mathbf{I}, c, m\}$, produced by a mass spectrometer. The spectrum consists of:

- $\mathbf{I} = \{(m/z_1, i_1), (m/z_2, i_2), \dots, (m/z_k, i_k)\}$, a set of k observed mass-to-charge and intensity pairs, filtered by a signal threshold.



992 Figure 7: Illustration of the disjoint expressive spaces between natural language (NL) and protein
993 sequences under fine-tuning. CoT reasoning capabilities learned in L_{NL} do not transfer into $L_{protein}$,
994 resulting in no shared reasoning ability.

- 995
- 996 • $c \in \mathbb{Z}^+$, the charge state of the precursor ion, and
- 997
- 998 • $m \in \mathbb{R}^+$, the total measured mass of the peptide.
- 999

1000 The objective is to **predict the underlying amino acid (protein) sequence** $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$,
1001 where each token $a_i \in \mathcal{V}_{protein} = \{20 \text{ amino acids tokens}\}$ belongs to the vocabulary of the protein
1002 language $L_{protein} = (\mathcal{G}_{protein}, \mathcal{V}_{protein})$. The mapping $\mathbf{H} \mapsto \mathbf{A}$ requires the model to reason over the
1003 structure and intensity patterns in \mathbf{I} while conforming to biochemical constraints imposed by c and
1004 m .

1005 *Note that, although we describe our method in the context of the protein sequence space, it applies to*
1006 *other biological sequence prediction tasks-including RNA, DNA, and synthetic polymers.*

1008 E WHY FINETUNING FAILS TO TRANSFER REASONING TO PROTEIN 1009 SEQUENCES

1010

1011 Let L_{NL} denote the natural language space (e.g., English), and $L_{protein}$ denote the protein language
1012 space, comprising amino acid tokens with limited compositional structure. Large language models
1013 (LLMs) such as LLaMA or GPT-4 are pretrained to reason over L_{NL} and can perform advanced
1014 tasks through CoT prompting—generating intermediate reasoning steps, storing temporary variables,
1015 performing error correction, and maintaining self-reflection, all in natural language.

1016

1017 In approaches like **ProLlama**, a pretrained natural language model is fine-tuned on protein sequences,
1018 introducing a new vocabulary $\mathcal{V}_{protein}$ and dataset distribution $\mathcal{S}_{protein} \subset \mathcal{X}_{protein}^*$. While the transformer
1019 architecture remains shared, the representations fine-tuned on proteins diverge significantly from
1020 those learned in natural language. As illustrated in Figure 7, the model’s learned expressiveness splits
1021 into two largely non-overlapping subspaces:

$$1022 \mathcal{S}_{NL} \cap \mathcal{S}_{protein} \approx \emptyset \tag{2}$$

1023

1024 This results in a model that performs well on surface-level protein generation but cannot leverage its
1025 natural language reasoning skills—because those abilities remain confined to the L_{NL} space and are
inaccessible in $L_{protein}$.

To understand this intuitively, consider a bilingual LLM pretrained on English. If we fine-tune it on Chinese texts without joint training or alignment, the model learns to generate Chinese tokens fluently, but cannot reason about Chinese using English thinking patterns. For example, tokens like “let me double check” or “I made a mistake” have no meaningful grounding in the Chinese token space unless the model was trained to use them cross-lingually. The model ends up with two islands of knowledge—English and Chinese—that cannot communicate or share reasoning capacity.

The same phenomenon occurs in protein modeling. ProLlama retains its CoT reasoning ability in English but cannot express or apply that reasoning in $\mathcal{L}_{\text{protein}}$. As a result, it cannot generate intermediate reasoning tokens like `<reflect>` or revise outputs mid-generation—capabilities that are essential for self-correction, debugging, or guided sequence exploration.

In contrast, our reflection-pretraining approach explicitly introduces auxiliary reasoning tokens $\mathcal{V}_{\text{reflect}}$ and trains the model to use them within the protein generation process. This expands the language from $\mathcal{L}_{\text{protein}}$ to an augmented form:

$$\mathcal{L}_{\text{protein}+} = (\mathcal{G}_{\text{protein}}, \mathcal{V}_{\text{protein}} \cup \mathcal{V}_{\text{reflect}}) \quad (3)$$

and correspondingly enlarges the expressive space from $\mathcal{S}_{\text{protein}}$ to $\mathcal{S}_{\text{protein}+}$, allowing the model to generate and interpret internal reasoning sequences natively in the protein domain:

$$|\mathcal{S}_{\text{protein}+}| \gg |\mathcal{S}_{\text{protein}}| \quad (4)$$

F REFLECTION TOKEN FOR DIFFERENT ERROR CORRECTION

In this section, we provide intuitive examples of how the proposed `<reflect>` token functions as a backtracking mechanism to enable error correction in biological sequence generation. The reflection token allows the model to explicitly mark a mistake and revise it, rather than committing to an incorrect sequence. We illustrate three common error types: **deletion**, **swapping**, and **modification**.

F.1 DELETION CORRECTION

Ground truth:

M--K--L--T--P

Wrong prediction:

M--K--T--P

(here residue L was mistakenly skipped)

With reflection:

M--K--<reflect>--insert (L) --T--P

The reflection token explicitly signals the error and recovers the missing L.

F.2 SWAPPING CORRECTION

Ground truth:

A--G--S--V

Wrong prediction:

A--S--G--V

(G and S are swapped)

With reflection:

A--S--<reflect>--swap (G, S) --G--V

The reflection token backtracks, flags the swap, and restores the correct order.

1080 F.3 MODIFICATION CORRECTION

1081

1082 **Ground truth:**

1083 M--R--Q--H

1084 **Wrong prediction:**

1085 M--R--E--H

1086 (Q was incorrectly predicted as E)

1087 **With reflection:**

1088 M--R--<reflect>--replace (E, Q) --Q--H

1090 The reflection token enables the model to mark the erroneous substitution and replace it with the
1091 correct amino acid.

1092

1093 F.4 SUMMARY

1094

1095 These examples show how the <reflect> token acts like a biological “backspace” or “edit marker,”
1096 allowing the model to recover from local mistakes. Unlike standard sequence generation—which
1097 commits to an error once produced—reflection-based generation provides a structured way to annotate,
1098 revise, and correct intermediate predictions. This ability is crucial for biological sequences, where
1099 even single-residue errors can have drastic structural or functional consequences.

1100

1101 G EXPERIMENTS

1102

1103 **Model Training.** We adopt the same architecture and optimization configuration as in prior work
1104 of training Transformer for De novo Peptide Sequencing. All inputs—including MS/MS peaks,
1105 precursor features, and amino acid tokens—are first embedded into a shared 400-dimensional latent
1106 space.1107 The model is built on a 9-layer Transformer, with each layer comprising 8 attention heads and a
1108 1024-dimensional feedforward network. This architecture forms the backbone for both the encoder
1109 and decoder components.1110 Training is conducted with a batch size of 1600 and an initial learning rate of 4e-4. The learning
1111 rate linearly warms up during the first epoch before decaying via a cosine schedule. Optimization is
1112 performed using AdamW (Kingma & Ba, 2014), and all models are trained for 30 epochs on eight
1113 A100 GPUs.1114 These hyperparameters—embedding dimension, transformer depth, attention head count, and learning
1115 rate policy—are kept fixed across all experiments unless explicitly stated. Additional implementation
1116 details can be found in our released codebase.1117 **Base Bio-inspired Model Designs.** We evaluate our approach against a range of bio-inspired
1118 peptide sequencing models, each of which incorporates domain-specific knowledge or architectural
1119 modifications tailored to biological data:

1120

- 1121
- **Peaks** (Ma et al., 2003) employs a protein database search strategy that combines tandem
1122 mass spectrometry with homology-based alignment. This approach excels in navigating
1123 ambiguity in peptide identification using prior biological information.
 - **DeepNovo** (Tran et al., 2017) combines convolutional and recurrent neural networks with
1124 peptide-specific preprocessing and adopts a species-wise cross-validation scheme.
 - **PointNovo** (Qiao et al., 2021) builds on point-cloud processing ideas to handle varied-
1125 resolution spectra without increasing model complexity, showcasing robustness through
1126 biologically-informed design.
 - **InstaNovo** (Eloff et al., 2023) introduces a diffusion model that improves performance
1127 through iterative refinement of predicted sequences.
 - **HelixNovo** (Yang et al., 2024) extends CasaNovo by modifying the MS/MS input to explic-
1128 itly encode complementary b/y ion relationships—information that is biologically grounded
1129 in peptide fragmentation rules.
- 1130
-
- 1131
-
- 1132
-
- 1133

1134 In contrast to these models, our reflection-pretrained Transformer introduces no biological heuristics
1135 or domain-specific preprocessing. Instead, it augments a standard model’s reasoning capacity via
1136 token-level reflection and self-correction. This purely cognitive enhancement allows the model to
1137 outperform bio-inspired baselines without incorporating any explicit biochemical priors.
1138

1139 **Analysis of Beam Size Influence.** We examined the effect of beam search width on the performance
1140 of the Reflection Pretrain model. As shown in Table 2, increasing the beam size consistently improves
1141 model performance across both Amino Acid (AA) Precision and Peptide Recall metrics. Specifically,
1142 AA Precision increased from 0.788 at beam size 1, to 0.797 at beam size 3, and further to 0.806 at
1143 beam size 5. This reflects an absolute improvement of 0.018, corresponding to a relative increase
1144 of approximately 2.28%. A similar trend was observed for Peptide Recall, which rose from 0.600
1145 (beam size 1) to 0.608 (beam size 3), and reached 0.617 at beam size 5—an absolute gain of 0.017
1146 and a relative improvement of approximately 2.83%. These results suggest that larger beam sizes
1147 enable the model to explore a broader hypothesis space during decoding, thereby reducing the risk
1148 of prematurely discarding promising sequence candidates. Nevertheless, the performance benefits
1149 must be balanced against the increased computational cost in both time and memory—an inherent
1150 trade-off in beam search.

1151 H THE USE OF LARGE LANGUAGE MODELS (LLMs)

1152
1153 Large Language Models (LLMs) served as assistive tools for improving the clarity and grammar of
1154 our academic prose. Specifically, we leveraged GPT-5 for drafting and refining sections such as the
1155 introduction and method. The authors retain full responsibility for all scientific content, including the
1156 conception of the research questions, methodological contributions, and the validation of experimental
1157 results.
1158

1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187