EVOLUTIONARY INFORMED TRANSLATION FROM PROTEIN TO NUCLEIC ACID SEQUENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

The choice of nucleic acid sequence for a given protein strongly impacts function and is species-specific. Current protein-to-nucleic-acid translation models treat sequences in isolation, often over-favoring frequently used codons. To generate more natural nucleic acid sequences, we introduce PINATA (Phylogenetically Informed protein to Nucleic Acid Translation with sequence Alignment), a model that leverages sequences from related species encoded in multiple sequence alignments (MSAs) to guide prediction. Our architecture leverages evolutionary context and learns species-and protein specific context to make biologically-informed predictions. Evaluated across 61,934 gene trees and 882 species, PINATA generates sequences that mirror the naturally-occurring sequence substantially better than existing language models. PINATA leverages MSA data to improve exact native codon recovery by 17.95 percentage points over existing language models, effectively preserves natural rare-codon nuances, and more closely matches native sequence properties. PINATA maintains strong predictive performance even with limited evolutionary data. While previous methods largely treated sequences from different species in isolation, our method leverages the evolutionary relationships between nucleic acids and proteins across species to create a more biologically-informed translation. We demonstrate that leveraging rich evolutionary data through an biologically-designed method strongly aids in translating protein sequences to nucleic acid sequences across species. We will make data and code publicly available upon publication.

1 INTRODUCTION

Biological information is encoded through large molecular sequences; these sequences and their interactions create biological language. Two fundamental sequence representations are nucleic acid sequences (DNA and RNA), which store and transmit key information using four nucleotide bases, and proteins, which are composed of 20 building blocks called amino acids and carry out a wide range of functions. These sequence representations can be viewed as distinct biological languages connected by the genetic code, which maps nucleotide triples, called codons, to amino acids during protein synthesis. While each codon specifies a single amino acid, the reverse translation from proteins to nucleic acids is not deterministic. There are $4^3 = 64$ nucleic acid codons but only 20 amino acids, so most amino acids are encoded by multiple synonymous codons, and an exponential number of synonymous nucleic acid sequences code for the same protein. This choice of nucleic acid sequence has consequential biological effects (Kudla et al., 2009; Boël et al., 2016; Varenne et al., 1984) and depends on both the properties of the protein and on the species (e.g., dogs, mice, *E. coli*). This intricate relationship means that even the same protein takes on a different nucleic acid sequence depending on the species in which it is expressed. This is especially relevant in bioengineering, where genes are frequently transferred across species (Demain & Vaishnav, 2009; Luo et al., 2015; Huo et al., 2019). Generating a native-like sequence could provide a starting point for multi-objective gene design that can be further optimized for specific qualities desired in bioengineering applications such as protein expression and solubility. We frame the problem as a translation spanning multiple axes of biological language: from protein language to nucleic acid language, across the languages of many species. This multi-faceted translation requires reasoning jointly over protein sequence, nucleic acid sequence, and species identity, offering a useful lens on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

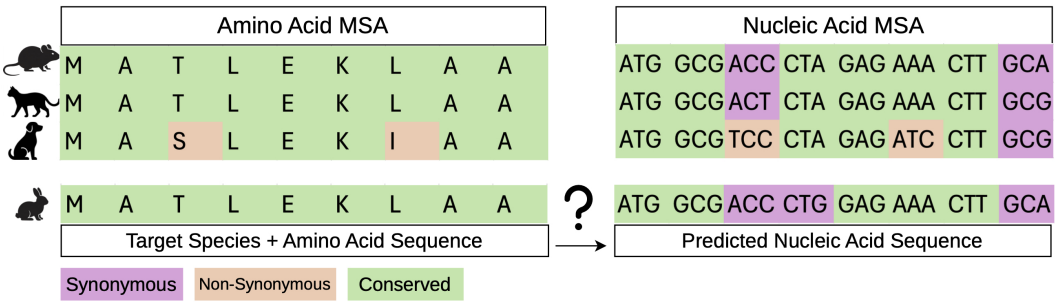


Figure 1: **Evolutionary Informed Translation from Protein to Nucleic Acid Sequence**. Protein sequences of multiple species (mouse, cat, dog, and rabbit) are aligned in the amino acid MSA (left). Each protein sequence is made up of amino acids (denoted with 1-letter codes) and each nucleic acid sequence is made up of a sequence of codons (denoted with 3-nucleotides per codon). The known nucleic acid sequences (mouse, cat, dog) are aligned in the nucleic acid MSA (right), by following the amino acid MSA. For the target rabbit protein sequence (MATLEKLAA), we predict the corresponding rabbit nucleic acid sequence (right). Conserved positions (green) are the same in both amino acid and nucleic acid choice. Synonymous (purple) positions are the same for the amino acid yet differ for the nucleic acid. Non-synonymous (orange) positions differ for both the amino acid and the nucleic acid. Our goal is to predict the nucleic acid sequence for the target protein and species by leveraging information in the MSAs.

multi-modal biological language modeling. With this motivation, we pose the central problem of our paper: how do we translate from protein language to nucleic acid language across species?

A key aspect of the problem is how nucleic acid languages across different species are related through evolution. The relatedness between nucleic acid sequences provides a key biological signal for understanding and recovering the natural sequence distributions across species. While each species has its own version of nucleic acid language, related species - such as chimps and humans - share high similarity in their nucleic acid sequences, with differences between the languages accumulating over time and giving rise to the species differences we see today (Dobzhansky, 1937). In particular, similarities between the languages can point to key functionality preserved for survival while differences can point to species-dependent features. To represent these evolutionary changes across species, biologists use multiple sequence alignments (MSAs) (Feng & Doolittle, 1987), where each row is a species and each column represents how a given position in the sequence changes across species (Figure 1). Specifically, paired MSAs, composed of protein and corresponding nucleic acid sequences, are constructed for homologous sequences (related sequences with the same or similar amino acid sequence from different species). For a given position in homologous sequences, three possibilities occur. First, the position can be conserved: the amino acid and the nucleic acid codon in both sequences are the same. Second, the position can have a synonymous change: the amino acid is the same but the codon is different. Third, the position can have a non-synonymous change: both the amino acid and the codon differ.

Critically, this evolutionary context is available for many proteins of interest. When translating a protein to a nucleic acid sequence of a particular species, we can leverage the homologous sequences from related species. This data can easily be found in databases such as Ensembl (Yates et al., 2026) or constructed with existing bioinformatics methods (Katoh & Standley, 2013). Certain species are more well-studied than others; leveraging MSA information enables context from well-studied species to be used when making predictions regarding data-scarce species. Current models for translating protein to nucleic acid sequences do not leverage MSA information nor do they directly account for evolution, treating species largely in isolation with the goal that a large enough model will learn patterns out of millions of sequences (Heuschkel et al., 2026; Fallahpour et al., 2025). Instead of modeling the problem as an evolutionary-informed translation across species, they represent the species as a context token, without capturing the evolutionary relationships between protein and nucleic acid sequences across species. Since these models are trained with the objective of selecting the correct codon, they tend to favor high-frequency codons, deviating from the species' natural sequence distribution.

In our work, we directly leverage the evolutionary relatedness between the nucleic acid languages of different species. By deeply integrating our model with the biological evolutionary process, we recover sequences much closer to the native (found-in-nature) sequence. We introduce PINATA, a method that efficiently injects evolutionary information at every position in the amino acid sequence and leverages attention to allow each token to attend to the context of the surrounding amino acids, the surrounding evolutionary context, and the species. This enables our model to uniquely adhere to the rich information encoded in homologous sequences while adapting to protein and species specific context. Our model creates more natural (closer to the native sequence), species-specific nucleic acid sequences across a wide variety of proteins and over 800 species. PINATA offers three key contributions:

- We leverage evolutionary MSA data from related species as key biological information for translating between protein and species-dependent nucleic acid sequences
- We deeply integrate the evolutionary information throughout our model and allow the model to learn species and protein specific context, creating a biologically-aligned design
- PINATA generates sequences that are substantially more similar to the native sequences than previous work, with a 17.95pp higher recovery of native codons compared to existing language models. We demonstrate that PINATA's improvements persist even when evolutionary data is scarce.

2 Related Work

Several language modeling approaches have been developed to convert a protein to a nucleic acid sequence of a given species. CodonTransformer (Fallahpour et al., 2025) utilizes an encoder-only architecture trained with a masked-language-modeling (MLM) objective. By adding a species encoding to amino acid-codon pair tokens, the model predicts masked tokens as the most likely codon given the amino acid sequence and species. SynCodonLM (Heuschkel et al., 2026) builds upon this approach by explicitly restricting the model to only predict synonymous codons. ppLMCo (Pathak & Lin, 2026) improves the efficiency of codon prediction with a linear layer on top of a pre-trained encoder. Very recently, CoFormer (Kim et al., 2026) integrates transformers and CNNs. GEMORNA (Zhang et al., 2025) is trained on mammalian genes, using an encoder-decoder architecture to predict the corresponding nucleic acid sequence, additionally generating untranslated regions outside of the coding sequence to influence expression. There are additional papers (Li et al., 2025; Rajbanshi & Guruacharya, 2025; Lin et al., 2026) that seek to optimize nucleic acid sequences for specific properties. These methods model species independently, without representing the evolutionary relationships between them. In our work, we demonstrate that evolution-based modeling renders recovery of native sequences.

A few models incorporate limited amounts of evolutionary data in their approach. Sidi et al. (2025), OrthologTransformer (Akiyama et al., 2026), and CodonT5 (Babjac & Emrich, 2024) give a single homologous protein as context. This does not, however, account for the large amount of evolutionary information encoded in dozens or hundreds of homologous sequences in the MSA. DeepCodon (Han et al., 2025) is trained solely on Enterobacteriaceae sequences, but uses homologous sequences to derive a binary signal for whether or not a position contains a rare codon. CodonTranslator (Chen et al., 2025) is a preprint which includes information about the relatedness between species in the species token; however, it does not include information about sequence relatedness through MSAs. Outside of our problem, there are modeling approaches for incorporating evolutionary information, such as encoding MSAs (Rao et al., 2021), predicting protein structure (Jumper et al., 2021), predicting variant effects (Riesselman et al., 2018; Frazer et al., 2021), and predicting protein fitness (Notin et al., 2022); however, these methods do not translate from protein to nucleic acid languages across species.

3 Methods

We design a bio-inspired model to deeply integrate evolutionary information into the translation between an amino acid sequence to the nucleic acid sequence across species. Our model is tailored to leverage the relatedness of sequences across species while enabling the model to capture nuance within the species and protein-specific context. An overview of our model is shown in Figure 2.

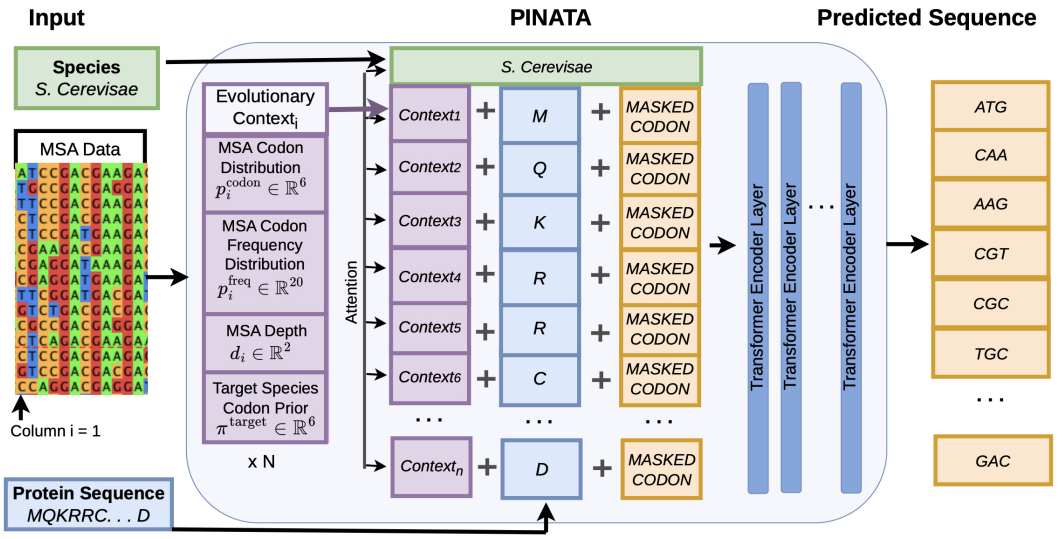


Figure 2: Overview of PINATA. PINATA extracts an evolutionary context vector from every position in the MSA. The context vector at every position is added to the amino acid and codon tokens. Attention occurs between the species token and the context-amino acid-codon tokens throughout a multi-layer transformer encoder. PINATA then decodes a synonymous nucleic acid sequence, taking into account the evolutionary context, protein context, and species-specific patterns. MSA Data visualized with JalView (Procter et al., 2021)

3.1 Evolutionary Context

We first design how to extract key information from the evolutionary context of the sequences from related species. To do this, we leverage MSA data encoding how nucleic acid sequences for a given protein change across species at each position of the protein. Details of this data are described in Section 4.1.

Each MSA consists of tens to hundreds of homologous sequences, amounting to a structure of size NxM, where N is the length of the target protein and M is the number of homologous sequences. We design a module to extract the most relevant biological information from this data into an efficient structure of 34 x N, where each column is a context vector of length 34.

There are two main pieces of evolutionary information that we extract at each position. First, for each target amino acid, we calculate the distribution of corresponding codons at that position in each homologous sequence. Since there is often significant similarity between nucleic acid sequences of related species, the corresponding homologous codons provides a strong signal. We define this information as MSA Codon Distribution, $\mathbf{p}_i^{\text{codon}} = \left(p_{i,1}^{\text{codon}}, \ldots, p_{i,6}^{\text{codon}}\right) \in \mathbb{R}^6$ where $p_{i,n}^{\text{codon}}$ represents the frequency of the n^{th} codon corresponding to the target amino acid a_i in column i. Each value is calculated across all non-target species in the MSA with the target amino acid at position i. As there are up to 6 codons per amino acid, the MSA Codon Distribution is in $\mathbb{R}^6$.

Each species has its own preferences for specific codons (Grantham et al., 1980). As a result, when a given amino acid position should be encoded by a rare codon, the codon actually used at that position in two different species will depend on the relative frequency, or preference, of that codon in each species' genome. To extract this codon frequency signal, we first compute the species-specific codon frequencies for all codons c and species s as $\pi_{s,c} = N_{s,c} / \sum_{c' \in \mathcal{C}(a)} N_{s,c'}$, where a is the amino acid encoded by c, $N_{s,c}$ is the number of occurrences of codon c in species s, and $\mathcal{C}(a)$ is the set of synonymous codons encoding a. Importantly, we only use training sequences when constructing this data. Then, for each position i in our target protein, we construct a histogram of the relative frequencies across corresponding codons. We define this histogram as the MSA Codon Frequency Distribution, $\mathbf{p}_i^{\text{freq}} = \left(p_{i,0}^{\text{freq}}, \ldots, p_{i,19}^{\text{freq}}\right) \in \mathbb{R}^{20}$. To construct this histogram, for each synonymous codon j at position i, we obtain the relative frequency $\pi_{s_j,c_{ij}}$ of its observed

codon c_{ij} in its species s_j. These values are assigned to 20 bins, of width 0.05 and normalized over the comparable homologous sequences. Thus, the value at each bin b for position i can be calculated as $p^{\text{freq}}_{i,b} = \frac{1}{|H_i|}\sum_{j\in H_i}\mathbb{I}\big[\min\big(\lfloor 20\pi_{s_j,c_{ij}}\rfloor, 19\big) = b\big], \quad b \in \{0,\ldots,19\}$, where H_i is the set of homologous sequences that contain a codon at position i encoding the same amino acid as the target.

We include two additional values as context. First, we recognize that not every target amino acid position has an equivalent number of synonymous codons in the homologous sequences (for example, some homologous sequences may have a gap or a different amino acid). We therefore include two values for how much support there is at a given position, i. We express this as $\mathbf{d}_i = \left[\frac{\log(1+n_i)}{\log(1+\max(1,N))}, \frac{n_i}{\max(1,N)}\right] \in \mathbb{R}^2$, where N is the total number of homologous sequences in the MSA after excluding the target species and n_i is the number of homologous sequences matching the target amino acid at that position.

Finally, we add species-specific context into each position to avoid over-fitting to homologous sequences. At each amino acid position, we provide the target species codon prior, $\pi_c = N_c / \sum_{c'\in\mathcal{C}(a)} N_{c'}$ where c is a given codon encoding amino acid a, N_c is the total number of codons c across all training sequences, and $\sum_{c'\in\mathcal{C}(a)} N_{c'}$ is the total number of codons encoding amino acid a across all training sequences of the target species. This is not evolutionary information and provides information regarding codon preferences to the model.

Placing all these contexts together, p_i^{codon}, p_i^{freq}, d_i, and π_c gives a 34-dimensional vector at each amino acid position i. We then project this context to the 768 dimensions of the amino acid and codon embeddings through a linear layer, followed by layer-normalization, and GELU (Hendrycks & Gimpel, 2016) activation.

3.2 Architecture

Our model efficiently injects the evolutionary context as a 34-feature vector described in Section 3.1 at every amino acid position. We add the projected biological context, the amino acid, and the codon embeddings together. This directly fuses the most relevant evolutionary context into each amino-acid/codon embedding.

We leverage attention to enable the model to learn nuanced protein and species-specific patterns. Attention within each transformer layer shares information between each joint evolutionary-context/amino-acid/codon embedding with the species embedding prepended to the sequence; allowing each position to account for its own and surrounding evolutionary context, surrounding amino acids, and species. The species vocabulary contains 882 species plus an unknown species token.

Our model uses a fixed sinusoidal positional encoding and passes the tokens through an encoder-only Transformer (Vaswani et al., 2017) trained with BERT-style (Devlin et al., 2019) masked language modeling (MLM). The model has 13 layers, a hidden dimension of 768, 12 self-attention heads of dimension 64, and a 2,944-dimensional GELU (Hendrycks & Gimpel, 2016) feed-forward layer. Each Transformer block uses pre-layer normalization and a dropout rate of 0.1. The output head produces six codon-slot logits. Slots not corresponding to a valid synonymous codon for the input amino acid are masked before decoding. The model contains approximately 90M parameters. Additional details are in Appendix A.1.

We limit each model input to a sequence length of 2,048. During training, longer sequences are truncated to the first 2,048 positions; during inference, they are divided into consecutive non-overlapping chunks of at most 2,048 positions, decoded independently with the species token repeated for each chunk, and concatenated to produce the sequence.

3.2.1 Training

During the first four epochs, we train the model to maximize the probability of native codons at masked positions given the target amino-acid sequence, the partially masked codon sequence, the evolutionary context, and the target species. We randomly replace approximately 15% of eligible

codon positions with a universal mask token and use the MLM cross-entropy loss

$$\mathcal{L}_{\text{naturalCE}} = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log p_\theta \left(c_i \mid \mathbf{a}, \widetilde{\mathbf{c}}, \mathbf{X}^{\text{context}}, s \right),$$

where $\mathcal{M}$ is the set of masked positions, c_i is the native ground-truth codon at position i, $\mathbf{a}$ is the target amino-acid sequence, $\widetilde{\mathbf{c}}$ is the codon sequence with positions in $\mathcal{M}$ replaced by mask tokens, $\mathbf{X}^{\text{context}}$ is the position-specific evolutionary context matrix, and s is the target species.

In the 5th epoch, we add an additional component to the loss, $\mathcal{L}_{\text{property}}$ to maximize how well the predicted sequence follows the natural properties of the ground truth sequence. We define $\mathcal{L}_{\text{property}}$ as a combination of terms which penalize according to the difference between properties of the predicted and natural sequence. We define $\mathcal{L}_{\text{property}} = \frac{1}{4}\left[\frac{|\Delta\text{GC}|}{5} + \frac{|\Delta\text{CFD}|}{5} + \frac{|\Delta\text{CSI}|}{0.1} + \frac{\text{softDTW}_{\text{norm}}}{10}\right]$ Here, ΔGC is the difference of the percentage of G and C content across all codons between the native and predicted sequence. ΔCFD (Fallahpour et al., 2025) is the difference in the amounts of rare codons, ΔCSI is the codon similarity index, measuring how much the sequence favors codons with a high frequency (Sabath et al., 2012), and normalized soft-DTW divergence (Cuturi & Blondel, 2017; Blondel et al., 2021) is a proxy for DTW distance (Giorgino, 2009) which measures how well the predicted sequence's codon frequency distribution mirrors that of the native sequence.

We combine the MLM loss with the property loss to create the final-epoch loss $\mathcal{L}_{\text{finalEpoch}} = \mathcal{L}_{\text{property}} + \mathcal{L}_{\text{naturalCE}}$. We use a 100% mask in the 5th epoch to better represent the downstream task and fine-tuned for up to one additional epoch and selected among quarter-epoch validation checkpoints.

4 Results

4.1 Data Collection

We collect MSAs from aligned protein gene trees in Ensembl Compara release 116 and Ensembl Genomes release 63 (Yates et al., 2026), encompassing Fungi (EFGT0), Metazoa (EMGT0), Vertebrates (ENSGT), Plants (EPIGT), and Protists (EPrGT). Notably, these gene trees each comprise of loosely related species (e.g., all fungi or all plants). We combine all of the MSAs from these gene trees into our dataset. Thus, our model works for many different types of species, and the MSA for each prediction is from a grouping of loosely related species.

We select one representative sequence per species within each gene tree and perform quality control, with details in the Appendix A.1.3. We retain trees containing at least 25 valid sequences from at least 25 distinct species. The resulting dataset consists of 61,934 MSAs and 5,933,724 protein-coding sequences across 882 distinct species. Dataset composition is reported in Table 1.

To limit leakage between training and evaluation sets, we cluster all protein sequences using MMseqs2 (Steinegger & Söding, 2017) at thresholds of at least 80% amino-acid identity and 80% coverage in both the query and target directions. Gene trees connected through any sequence cluster are treated as a connected component. Components are assigned to training, standard validation, and standard test sets, targeting a 90%/5%/5% distribution by sequence count. For the standard set, no member of the target MSA appears during training, and members of the target MSA from non-target species provide evolutionary context at inference.

We also construct a within-tree test set composed of proteins with homologous sequences present during training by holding out approximately 5% of sequences from train-set MSAs. The purpose of this test set is to measure how our model compares even when all models have seen the homologous sequences during training. We randomly generate 3 different splits of the dataset.

We demonstrate that MSAs from loosely related species encode rich biological information due to conservation. We measure synonymous-codon conservation by comparing codon choices among homologous residues encoding the same amino acid and contrast the observed agreement with the agreement expected from species-specific codon usage. We find that across every division of our dataset, codon conservation produces a strong signal beyond what would be expected by random chance, demonstrated in Table 1.

Table 1: Dataset distribution and synonymous-codon conservation

Division	Gene trees	Sequences (%)	Observed conservation (%)	Null conservation (%)
Fungi	16,240	1,650,655 (27.82%)	39.98	34.27
Metazoa	8,141	422,104 (7.11%)	41.63	33.90
Vertebrates	18,345	2,446,863 (41.24%)	60.70	35.68
Plants	14,242	1,138,013 (19.18%)	57.59	33.91
Protists	4,966	276,089 (4.65%)	47.06	37.62
Total	61,934	5,933,724 (100.00%)	50.08	34.91

To evaluate out-of-distribution generalization, we construct 1,000 bacterial codon MSAs from GTDB release R232 (Parks et al., 2026). We select 500 high-quality representative genomes spanning 10 bacterial phyla, predict protein-coding genes with Prodigal (Hyatt et al., 2010), and infer single-copy orthogroups separately within each phylum using OrthoFinder (Emms & Kelly, 2019). Orthogroups containing at least 25 species are aligned at the amino-acid level with MAFFT (Katoh & Standley, 2013) and back-translated to codon alignments using the corresponding coding sequences. We select one target from each of 1,000 distinct orthogroups and exclude targets with at least 80% sequence identity and 80% bidirectional coverage to any training protein using MMSeqs2 (Steinegger & Söding, 2017) from the training.

4.2 Evaluation Metrics

We evaluate how well our generated codon sequences represent the distribution found in the native sequence through the following metrics:

- Exact Codon Match: The percentage of codon positions where the predicted codon is identical to the native codon
- GC Content: The percentage of G and C nucleotides in the predicted codon sequence
- Codon Similarity Index (CSI) (Sabath et al., 2012): The geometric mean of positive relative codon adaptiveness values. We use the frequency of codons by species in our full dataset as the ground-truth frequencies. CSI demonstrates how the sequence favors codons that are more frequently used by the species.
- Codon Frequency Deviation (CFD): Calculated as in Fallahpour et al. (2025): the percentage of predicted codons with relative adaptiveness below 0.3.
- Minimum Free Energy (MFE): The predicted mRNA folding energy, calculated using ViennaRNA (Lorenz et al., 2011). Due to the computational time needed to compute MFE, MFE was computed on a random subset of 50,000 sequences with a sequence length limit of 2,048. The same subset was used across all models.
- DTW Distance: As in Fallahpour et al. (2025), we follow the algorithm in Giorgino (2009), the dynamic time warping distance between the predicted and native sliding window of %MinMax (Clarke IV & Clark, 2008) codon adaptation profiles (18-codon window). DTW Distance measures how well the predicted sequence's codon frequency distribution mirrors that of the native sequence.

To demonstrate how well the predictive sequence matches the properties of the native sequence, we present GC content, CSI, CFD, and MFE as the sequence-level absolute difference between the predicted and native sequence, macro-averaged over sequences.

4.3 Model Performance

In our experiments, we answer the following questions: (1) Does PINATA generate more native-like sequences than state-of-the-art models? (2) What is the cause behind PINATA's improvements? (3) Do PINATA's improvements persist even when all models have seen the homologous sequences during training, (4) How does PINATA perform when the amount of MSA data is scarce, and (5) How does PINATA perform out-of-distribution?

Table 2: Model Performance (Standard Dataset)

Model	Codon match (%) ↑	|ΔGC| (pp) ↓	|ΔCFD| (pp) ↓	|ΔCSI| ↓	Normalized DTW ↓	|ΔMFE| (kcal/mol) ↓
PINATA (Ours)	**67.94 ± 0.06**	**1.426 ± 0.004**	**0.962 ± 0.003**	**0.0209 ± 0.0001**	**4.538 ± 0.004**	**17.95 ± 0.12**
CodonTransformer	49.99 ± 0.09	6.859 ± 0.478	3.218 ± 0.076	0.1189 ± 0.0003	14.202 ± 0.423	73.39 ± 4.78
SynCodonLM	43.21 ± 1.30	11.055 ± 1.377	3.812 ± 0.353	0.1237 ± 0.0152	19.089 ± 2.492	104.00 ± 19.95
PINATA (no MSA ablation)	50.35 ± 0.08	2.843 ± 0.039	1.502 ± 0.010	0.0407 ± 0.0004	6.387 ± 0.007	31.36 ± 0.89
MSA signal only (no learning)	63.01 ± 0.01	2.698 ± 0.001	1.436 ± 0.006	0.0353 ± 0.0001	5.361 ± 0.009	24.91 ± 0.12

Values are mean ± sample standard deviation over 3 runs. Metrics were calculated over approximately 296,000 sequences and MFE is evaluated on each run's shared frozen 50,000-sequence cohort. SynCodonLM calculates its evaluation metrics over a subset of 50,000 sequences due to computational expense.

Table 3: Model Performance (Within-tree dataset)

Model	Codon match (%) ↑	|ΔGC| (pp) ↓	|ΔCFD| (pp) ↓	|ΔCSI| ↓	Normalized DTW ↓	|ΔMFE| (kcal/mol) ↓
PINATA (Ours)	**65.77 ± 0.06**	**1.448 ± 0.016**	**0.964 ± 0.008**	**0.0217 ± 0.0000**	**4.600 ± 0.007**	**21.34 ± 0.17**
CodonTransformer	49.97 ± 0.48	6.731 ± 0.602	3.197 ± 0.059	0.1110 ± 0.0005	12.366 ± 0.427	86.70 ± 6.58
SynCodonLM	42.78 ± 1.04	11.153 ± 1.431	4.073 ± 0.383	0.1247 ± 0.0131	18.167 ± 2.335	122.97 ± 21.78
PINATA (no MSA ablation)	50.63 ± 0.11	2.616 ± 0.022	1.395 ± 0.009	0.0382 ± 0.0002	5.954 ± 0.013	34.82 ± 1.51
MSA signal only (no learning)	58.49 ± 0.01	3.737 ± 0.011	2.081 ± 0.006	0.0487 ± 0.0001	6.348 ± 0.008	37.24 ± 0.08

Values are mean ± sample standard deviation over 3 runs. Metrics were calculated over approximately 269,000 sequences and MFE is evaluated on each run's shared frozen 50,000-sequence cohort. SynCodonLM calculates its evaluation metrics over a subset of 10,000 sequences due to computational expense.

We compare our model against two state-of-the art language modeling methods for species-dependent conversion of protein sequences to nucleic acid sequences: CodonTransformer (Fallahpour et al., 2025) and SynCodonLM (Heuschkel et al., 2026). For a controlled comparison, we train both methods from scratch on the three same training splits used by our model, using the same five-epoch training budget and selecting checkpoints using the standard dataset validation performance. For SynCodonLM, we are only able to evaluate 50,000 sequences with length no greater than 2,048 on the standard dataset and 10,000 sequences with length no greater than 2,048 on the within-tree set per run due to the computational time required by its inference: these results are included in the main tables with a comparison against all models on the exact subset of sequences shown in the Appendix A.4. We adhere as closely as possible to each method's original implementation and training procedure, with complete details provided in the Appendix A.5.

To answer question 1, we test our model on the set of held-out gene trees. PINATA outperforms the leading language model at predicting native-like sequences across all metrics, including a 17.95 percentage point higher codon match and 68.0% lower DTW (Table 2). These results demonstrate that PINATA generates sequences with significantly higher similarity to the native sequence than previous language models.

To demonstrate evolutionary context as the source of PINATA's improvement (question 2), we ablate all MSA-derived features (the homolog codon distribution, homolog frequency histogram, and MSA depth) at both training and inference. Removing MSA data degrades performance on all metrics (Table 2), confirming that evolutionary context is central to the model. We implement a simple heuristic that, at each position, selects the most frequently observed synonymous codon among homologs encoding the target amino acid, falling back to the target species' training-derived codon frequency prior when no such homolog exists. This heuristic, shown as MSA signal only (no learning) in the table, alone outperforms SOTA language models on every metric, improving codon match by 13.02pp. Learning nuanced context contributes further gains: PINATA adds a further 4.93pp gain in codon match and improvements on every metric over the heuristic, demonstrating the importance of leveraging MSA information with context-specific learning.

To show PINATA's improvement even when all models have seen the homologous sequences during training (question 3), we use a within-tree test-set. We hold out approximately 5% of sequences from each training-set gene tree for the test set and place the homologous sequences in the training set. All models see the homologous sequences during training. In this setting, PINATA recovers a substantially more natural distribution than leading language models, shown in Table 3 with improvements similar to those in the standard data set, showing that our methodology for incorporating evolutionary context from homologous sequences at inference substantially boosts recovery of the native nucleic acid sequence.

Table 4: Model Performance with Reduced MSA Context

Closest homologs removed (%)	Codon match (%) ↑	|ΔGC| (pp) ↓	|ΔCFD| (pp) ↓	|ΔCSI| ↓	Normalized DTW ↓	|ΔMFE| (kcal/mol) ↓
10	62.86	1.692	1.100	0.0240	4.976	20.63
25	60.12	1.856	1.178	0.0255	5.231	22.16
50	57.66	2.014	1.267	0.0270	5.446	23.33
75	54.48	2.274	1.386	0.0294	5.729	25.60
90	52.00	2.510	1.507	0.0318	6.010	27.61
100	45.89	4.161	2.243	0.0436	7.784	39.68
No-MSA ablation	50.38	2.828	1.504	0.0406	6.383	30.96

Homologs removed in order of increasing target-to-homolog patristic distance, so the phylogenetically closest sequences were removed first. Metrics were calculated over 296,703 sequences and MFE uses the same frozen cohort of 50,000 sequences across one run

Table 5: Bacterial Out-of-Distribution Performance

Model	Codon match (%) ↑	|ΔGC| (pp) ↓	|ΔCFD| (pp) ↓	|ΔCSI| ↓	Normalized DTW ↓	|ΔMFE| (kcal/mol) ↓
PINATA (Ours)	**48.13**	**4.260**	**5.550**	**0.0887**	**8.219**	**28.97**
CodonTransformer	43.09	8.980	8.252	0.1308	19.875	55.44
SynCodonLM	44.60	13.382	11.791	0.1885	29.405	77.41
PINATA (no MSA)	38.89	6.247	14.060	0.2111	16.801	46.92
MSA signal only (no learning)	47.39	5.086	5.766	0.0946	8.753	36.38

Results are for 1,000 held-out bacterial targets across one run

We next evaluate PINATA when MSA data is scarce (question 4), progressively removing the 10%, 25%, 50%, 75%, 90%, and 100% most closely related homologs (Table 4). PINATA outperforms the no-MSA ablation with up to 90% of homologs removed. With the MSA removed entirely, it falls below that ablation, since it was never trained without evolutionary context.

In Table 5, we demonstrate how PINATA performs out of distribution (question 5) with a bacteria dataset. No models were trained on bacteria. To minimize information about the unseen species, we gave only a uniform prior instead of the target species codon frequency prior. For the 1,000 bacterial sequences predicted, PINATA exhibited the best performance. Inclusion of MSA information improves model accuracy even out of distribution.

5 Conclusions and Future Work

We develop an evolution-aware model translating protein sequences to nucleic acid sequences across species. Due to its grounding in biological knowledge, PINATA generates nucleic acid sequences that are substantially closer to the native sequence than traditional language models, with 17.95pp more matching codons. PINATA is a biologically-aware model which can translate across multiple biological modalities: amino acids, nucleic acids, and species. We demonstrate that providing evolutionary context provides a strong foundation for our model's predictions, and the flexibility of the model captures nuanced dependencies between the sequence, the evolutionary context, and the species context.

Future work includes adapting this model for multi-objective gene-transfer design. Gene design objectives are complex and application-dependent. Our model generates a native-like nucleic acid sequence for a given protein and species; future work can explore using these sequences as a starting point. A natural extension would be to leverage fine-tuning or reinforcement learning to optimize the generated sequences for specific functional properties, including solubility and expression, with wet-lab validation.

More broadly, our work emphasizes that explicitly encoding biological knowledge, rather than relying on scale to recover it, is a productive design principle for biological language models. Future work could incorporate biological signals beyond evolution and include additional modalities.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, create software code, assist in data formatting, to edit the research paper to improve readability, and to assist in drafting the abstract and appendix. In addition to a manual literature review, we used generative AI tools to identify relevant literature and we verified the accuracy of all citations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide details of the methodology for reproducibility in our methods and our appendix. Our data is derived from publicly available datasets;we describe our procedure to obtain the data in the appendix. Additionally, we will release a publicly available codebase upon publication.

REFERENCES

Manato Akiyama, Motohiko Tashiro, Ying Huang, Mika Uehara, Taiki Kanzaki, Mitsuhiro Itaya, Masakazu Kataoka, Kenji Miyamoto, and Yasubumi Sakakibara. Cross-species gene redesign leveraging ortholog information and generative modeling. *Nature Communications*, 17(1):2120, 2026.

Ashley Babjac and Scott J Emrich. CodonT5: A multi-task codon language model to perform generative codon optimization. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 12–18. IEEE, 2024.

Mathieu Blondel, Arthur Mensch, and Jean-Philippe Vert. Differentiable divergences between time series. In *International Conference on Artificial Intelligence and Statistics*, pp. 3853–3861. PMLR, 2021.

Grégory Boël, Reka Letso, Helen Neely, W Nicholson Price, Kam-Ho Wong, Min Su, Jon D Luff, Mayank Valecha, John K Everett, Thomas B Acton, et al. Codon influence on protein expression in E. coli correlates with mRNA levels. *Nature*, 529(7586):358–363, 2016.

Yanshuo Chen, Yuming Zhang, Joshua Li, Boxue Tian, and Heng Huang. CodonTranslator: a conditional codon language model for codon optimization across life domains. *bioRxiv*, pp. 2025–11, 2025.

Thomas F Clarke IV and Patricia L Clark. Rare codons cluster. *PloS one*, 3(10):e3412, 2008.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Arnold L Demain and Preeti Vaishnav. Production of recombinant proteins by microbes and higher organisms. *Biotechnology advances*, 27(3):297–306, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Theodosius Dobzhansky. *Genetics and the origin of species*. Number 11. Columbia university press, 1937.

David M Emms and Steven Kelly. OrthoFinder: phylogenetic orthology inference for comparative genomics. *Genome biology*, 20(1):238, 2019.

Adibvafa Fallahpour, Vincent Gureghian, Guillaume J Filion, Ariel B Lindner, and Amir Pandi. CodonTransformer: a multispecies codon optimizer using context-aware neural networks. *Nature Communications*, 16(1):3205, 2025.

Da-Fei Feng and Russell F Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of molecular evolution*, 25(4):351–360, 1987.

Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

Toni Giorgino. Computing and visualizing dynamic time warping alignments in R: the dtw package. *Journal of statistical Software*, 31:1–24, 2009.

R Grantham, C Gautier, M Gouy, R Mercier, and A Pavé. Codon catalog usage and the genome hypothesis. *Nucleic acids research*, 8(1):197–197, 1980.

Xudong Han, Xiaotong Shao, Liu Shuo, Zhenkun Shi, Rong Huang, Huanyu Chu, Hejian Zhang, Ruoyu Wang, Haoran Li, Xiaoping Liao, et al. DeepCodon: A deep learning codon-optimization model to enhance protein expression. *BioDesign Research*, pp. 100042, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

James Heuschkel, Laura Kingsley, Noah Pefaur, Andrew Nixon, and Steven Cramer. Advancing codon language modeling with synonymous codon constrained masking. *Nucleic Acids Research*, 54(5):gkag166, 2026.

Liujie Huo, Joachim J Hug, Chengzhang Fu, Xiaoying Bian, Youming Zhang, and Rolf Müller. Heterologous expression of bacterial natural product biosynthetic pathways. *Natural product reports*, 36(10):1412–1436, 2019.

Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, 11(1):119, 2010.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Kazutaka Katoh and Daron M Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.

Juseong Kim, Jeongmu Kim, Jae-Wook Lee, Qian Qi, Caroline Danehy, Ho Young Kang, Yong Cheng, and Giltae Song. Enhancing protein expression in humans through codon optimization with transformer and contrastive learning. *Molecular Therapy Nucleic Acids*, 37(3), 2026.

Grzegorz Kudla, Andrew W Murray, David Tollervey, and Joshua B Plotkin. Coding-sequence determinants of gene expression in escherichia coli. *Science*, 324(5924):255–258, 2009.

Yupeng Li, Fan Wang, Jiaqi Yang, Zirong Han, Linfeng Chen, Wenbing Jiang, Hao Zhou, Tong Li, Zehua Tang, Jianxiang Deng, et al. Deep generative optimization of mRNA codon sequences for enhanced mRNA translation and therapeutic efficacy. *Nature communications*, 16(1):9957, 2025.

Shenggeng Lin, Hong Tan, Keyao Wang, Ruixuan Wang, Hongxia Wang, Tong Zhu, and Yi Xiong. RNARL: reinforcement learning-driven unified generative framework for multi-objective RNA codon design. *Genome Biology*, 2026.

Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for molecular biology*, 6(1):26, 2011.

Yunzi Luo, Bing-Zhi Li, Duo Liu, Lu Zhang, Yan Chen, Bin Jia, Bo-Xuan Zeng, Huimin Zhao, and Ying-Jin Yuan. Engineered biosynthesis of natural products in heterologous hosts. *Chemical Society Reviews*, 44(15):5265–5290, 2015.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022.

Donovan H Parks, Pierre-Alain Chaumeil, Aaron J Mussig, Christian Rinke, Maria Chuvochina, and Philip Hugenholtz. GTDB release 10: a complete and systematic taxonomy for 715 230 bacterial and 17 245 archaeal genomes. *Nucleic Acids Research*, 54(D1):D743–D754, 2026.

Shashank Pathak and Guohui Lin. ppLM-CO: Parameter-efficient codon optimization with frozen pre-trained protein language model and guaranteed translation fidelity. In *Proceedings of the 17th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–6, 2026.

James B. Procter, G. Mungo Carstairs, Ben Soares, Kira Mourão, T. Charles Ofoegbu, Daniel Barton, Lauren Lui, Anne Menard, Natasha Sherstnev, David Roldan-Martinez, Suzanne Duce, David M. A. Martin, and Geoffrey J. Barton. *Alignment of Biological Sequences with Jalview*, pp. 203–224. Springer US, New York, NY, 2021. ISBN 978-1-0716-1036-7. doi: 10.1007/978-1-0716-1036-7_13. URL https://doi.org/10.1007/978-1-0716-1036-7_13.

Binita Rajbanshi and Anuj Guruacharya. codonGPT: reinforcement learning on a generative language model enables scalable mRNA design. *Nucleic Acids Research*, 53(22):gkaf1345, 2025.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. In *International conference on machine learning*, pp. 8844–8856. PMLR, 2021.

Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Niv Sabath, Andreas Wagner, and David Karlin. Evolution of viral proteins originated de novo by overprinting. *Molecular biology and evolution*, 29(12):3767–3780, 2012.

Eric W Sayers, Jeffrey Beck, Evan E Bolton, J Rodney Brister, Jessica Chan, Ryan Connor, Michael Feldgarden, Anna M Fine, Kathryn Funk, Jinna Hoffman, et al. Database resources of the national center for biotechnology information in 2025. *Nucleic acids research*, 53(D1):D20–D29, 2025.

Conrad L Schoch, Stacy Ciufo, Mikhail Domrachev, Carol L Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard Mcveigh, Kathleen O'Neill, Barbara Robbertse, et al. NCBI Taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062, 2020.

Tomer Sidi, Shir Bahiri-Elitzur, Tamir Tuller, and Rachel Kolodny. Predicting gene sequences with AI to study codon usage patterns. *Proceedings of the National Academy of Sciences*, 122(1): e2410003121, 2025.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Stanislas Varenne, Jean Buc, Roland Lloubes, and Claude Lazdunski. Translation is a non-uniform process: effect of tRNA availability on the rate of elongation of nascent polypeptide chains. *Journal of molecular biology*, 180(3):549–576, 1984.

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

Andrew D Yates, Olanrewaju Austine-Orimoloye, Andrey G Azov, Matthieu Barba, If Barnes, Vianey Paola Barrera-Enriquez, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, et al. Ensembl 2026. *Nucleic Acids Research*, 54(D1):D1053–D1060, 2026.

He Zhang, Hailong Liu, Yushan Xu, Haoran Huang, Yiming Liu, Jia Wang, Yan Qin, Haiyan Wang, Lili Ma, Zhiyuan Xun, et al. Deep generative models design mRNA sequences with enhanced translational capacity and stability. *Science*, 390(6773):eadr8470, 2025.

A Appendix

A.1 Dataset Construction and Quality Control

A.1.1 Ensembl gene trees and sequence filtering

We constructed the primary dataset from the default aligned protein gene-tree collections in Ensembl Compara release 116 and Ensembl Genomes release 63 (Yates et al., 2026). We considered protein members with canonical gene membership, a stable tree identifier, and an associated alignment. Protein sequences were obtained from the Compara sequence tables, coding sequences (CDSs) from the corresponding member-sequence tables, and alignment coordinates from each member's CIGAR string.

We retained only CDSs containing the canonical bases `A`, `C`, `G`, and `T`. Each CDS was required to translate under the standard nuclear genetic code to its recorded peptide, to contain no internal stop codon, and to be consistent with its alignment CIGAR. A terminal stop was retained as a separate prediction position when present. Species identifiers were normalized to their nearest NCBI (Schoch et al., 2020) species-rank ancestor, resulting in 882 normalized species. When a gene tree contained more than one valid sequence from the same normalized species, we retained the lexicographically smallest division/genome/protein identifier. We reapplied the requirement of at least 25 valid sequences from at least 25 distinct normalized species after all quality-control and representative-selection steps.

The protein alignment supplied by Ensembl defines the correspondence between rows. We back-translated each aligned amino-acid residue to its native CDS codon; the nucleotide alignment was therefore not constructed independently. For a particular target, columns in which the target was a gap were omitted, so the resulting feature sequence has one position per target codon.

A.1.2 Data Splits

Table 6 gives the realized split sizes.

Table 6: Realized sequence and gene-tree counts for the three reported repeated splits.

	Training		Standard set validation		Standard test		Within-tree test	
Run	Sequences	Trees	Sequences	Trees	Sequences	Trees	Sequences	Trees
1	5,070,637	48,510	296,679	6,716	296,703	6,708	269,705	48,510
2	5,071,532	48,507	296,685	6,710	296,685	6,717	268,822	48,507
3	5,071,019	48,513	296,691	6,709	296,666	6,712	269,348	48,513

A.1.3 Synonymous-codon conservation analysis

For the conservation analysis, we excluded gaps, stop codons, methionine, and tryptophan. Within each alignment column, residues were partitioned by amino acid, and an amino-acid group contributed only when at least two comparable residues were present. Every unordered pair in an eligible group contributed to the denominator, and a pair was counted as conserved when both members used the same synonymous codon. If multiple amino-acid groups were eligible in the same column, their matching-pair and total-pair counts were pooled before calculating the column score. Reported conservation is the equal-weight mean of these per-column scores.

The null expectation preserves the gene tree, alignment column, species, amino acids, and gaps. For species s and t in an amino-acid group, the expected probability of a synonymous match is calculated analytically as

$$P_0(\text{match} \mid a, s, t) = \sum_{c \in \mathcal{C}(a)} \pi_{s,c}\pi_{t,c},$$

where the species-specific distributions for this descriptive analysis were calculated over the complete dataset. Thus, the null does not assume a uniform codon distribution. Confidence intervals were obtained from 2,000 gene-tree bootstrap replicates stratified by Ensembl division.

A.1.4 BACTERIAL OUT-OF-DISTRIBUTION DATASET

We selected bacterial genomes from GTDB release R232 (Parks et al., 2026). Eligible assemblies were GTDB species representatives with estimated completeness of at least 90%, contamination of at most 5%, and NCBI (Sayers et al., 2025) translation table 11. We ranked phyla by the number of eligible genera and selected the ten largest phyla, breaking ties by phylum name. Within each phylum, genomes were first ranked within genus by isolate status, completeness, contamination, assembly level, and accession. A deterministic SHA-256 ordering with seed 232001 then selected 50 distinct genera per phylum, giving 500 genomes.

Only the selected assemblies were downloaded. Protein-coding genes were called with Prodigal 2.6.3 (Hyatt et al., 2010) in single-genome mode with translation table 11. Partial genes and CDSs that failed canonical translation to their reported proteins were removed. OrthoFinder (Emms & Kelly, 2019) was run separately within each phylum using `diamond_ultra_sens`. We retained orthogroups with exactly one protein per occupied genome and at least 25 occupied species. Each candidate group was aligned at the protein level with MAFFT (Katoh & Standley, 2013), after which the corresponding CDSs were back-translated through the amino-acid alignment.

Candidate targets were divided into length strata of at most 250, 251–500, and more than 500 amino acids. We selected 100 primary orthogroups per phylum and retained up to 20 reserves per phylum. Targets were chosen using a seeded ordering that balanced assignments across genomes, with at most five targets per genome. Each selected target came from a different orthogroup. Targets were limited to 2,047 sense amino acids so that the appended terminal-stop position produced at most 2,048 codons. The resulting benchmark contains exactly 1,000 targets, with 100 from each of ten phyla. The target is excluded from its own context, leaving 24–49 homolog rows. At a target position, a homolog contributes only when its aligned residue encodes the same amino acid; gaps and different amino acids do not contribute.

To exclude overlap with model training, each candidate target was searched against the 5,070,637 Run 1 training proteins using MMseqs2 (Steinegger & Söding, 2017) with `--min-seq-id 0.8 -c 0.8 --cov-mode 0 -s 7.5`. A target was rejected if any training hit had at least 80% identity, 80% query coverage, and 80% target coverage; a reserve from the same phylum and length stratum was then used.

A.2 ARCHITECTURE DETAILS

The amino-acid vocabulary contains the 20 standard amino acids, a terminal- stop state, and padding. The codon-input vocabulary contains the 64 native codons, one universal mask token, and padding. For each biological position, we sum its amino-acid embedding, codon-or-mask embedding, projected 34-dimensional context, and fixed sinusoidal positional encoding. The context projection is a linear map from 34 to 768 dimensions followed by layer normalization, GELU, and dropout with probability 0.1. A learned species embedding is prepended to the biological sequence. Each run's vocabulary is constructed from training species, with identifier 0 reserved for an unknown species; the reported runs contain 882 training-species tokens plus the unknown token.

The encoder contains 13 pre-layer-normalized Transformer blocks, hidden width 768, 12 attention heads of width 64, and a 2,944-dimensional GELU (Hendrycks & Gimpel, 2016) feed-forward sublayer. Dropout is 0.1. The implementation uses FlashAttention 2 (Dao, 2024)(version 2.8.3 in the recorded runs) and a final layer normalization. The six-dimensional output head represents amino-acid-specific synonymous-codon slots. Before both loss calculation and decoding, invalid slots are assigned a large negative logit. The model contains 90,365,062 trainable parameters.

A.3 TRAINING

During the first four epochs, each eligible position is independently selected with probability 0.15 and is replaced with the universal codon-mask token, with training details in Table 7. Sequences longer than 2,048 biological positions are truncated to their first 2,048 positions during training.

At inference, all target codons are replaced with the universal mask token in a single non-autoregressive pass. The synonymous codon with the largest valid logit is selected deterministically at every position. Sequences with length greater than 2,048 are partitioned into consecutive, non-

Table 7: Training configuration for PINATA.

Setting	Epochs 1–4	Sequence-property fine-tuning
Target masking	15%	100%
Loss	Masked natural CE	Natural CE $+\mathcal{L}_{\text{property}}$
Optimizer	AdamW	AdamW
Peak learning rate	2×10^{-4}	2×10^{-6}
Schedule	Constant	250-step warmup, cosine decay
Minimum learning rate	2×10^{-4}	2×10^{-7}
Weight decay	0.01	0.01
Gradient clipping	1.0	1.0
Precision	FP16 mixed	FP16 mixed
Maximum biological positions	2,048	2,048
Padded tokens per optimizer update	approximately 98,304	approximately 98,304
Checkpoint selection	Completed epoch 4	Joint fully masked validation objective

Table 8: Comparison on a Shared 50,000-Target Cohort (standard dataset)

Model	Codon match (%) ↑	$\lvert\Delta\text{GC}\rvert$ (pp) ↓	$\lvert\Delta\text{CFD}\rvert$ (pp) ↓	$\lvert\Delta\text{CSI}\rvert$ ↓	Normalized DTW ↓	$\lvert\Delta\text{MFE}\rvert$ (kcal/mol) ↓
PINATA (Ours)	**68.01 ± 0.06**	**1.422 ± 0.007**	**0.961 ± 0.008**	**0.0210 ± 0.0001**	**4.533 ± 0.010**	**17.95 ± 0.12**
CodonTransformer	50.03 ± 0.10	6.852 ± 0.478	3.210 ± 0.065	0.1189 ± 0.0000	14.240 ± 0.429	73.39 ± 4.78
SynCodonLM	43.21 ± 1.30	11.055 ± 1.377	3.812 ± 0.353	0.1237 ± 0.0152	19.089 ± 2.492	104.00 ± 19.95
PINATA (no MSA ablation)	50.37 ± 0.06	2.848 ± 0.044	1.504 ± 0.007	0.0408 ± 0.0006	6.400 ± 0.031	31.36 ± 0.89
MSA signal only (no learning)	63.07 ± 0.05	2.696 ± 0.010	1.426 ± 0.017	0.0352 ± 0.0003	5.360 ± 0.026	24.91 ± 0.12

overlapping chunks of at most 2,048 positions. The species token is repeated, positional indices restart, and each chunk is decoded independently before concatenation.

A.4 Results on sets matching SynCodonLM

Since we could only run 50,000 sequences in the standard test set and 10,000 sequences in the within-tree test set with length no greater than 2,048 on SynCodonLM (Heuschkel et al., 2026) due to long inference time, we additionally provide results of all models on only these 50,000 and 10,000 sequences, respectively, in Tables 8 and 9.

A.5 Comparison Methods

A.5.1 CodonTransformer

We followed the official CodonTransformer implementation (Fallahpour et al., 2025) at Git commit `4a447b01dab860feb81b647ff1ff88ad598517f4`. We adapted its data-loading pipeline to train on our three dataset splits, constructed the species vocabulary using only the training species from each split, and used deterministic validation loss for checkpoint selection. To avoid discarding positions from sequences longer than the model context window, we divided them into consecutive chunks of at most 2,046 codons and concatenated their predictions.

A.5.2 SynCodonLM

We followed the official SynCodonLM implementation (Heuschkel et al., 2026) at Git commit `9a773d9e44d1ea6e94e872cbabf41161edb54a28`. We adapted its data-loading pipeline to train independently on our three dataset splits, divided sequences longer than its context window into consecutive non-overlapping windows, and used deterministic validation loss for checkpoint selection. For a matched training budget, we trained for the same five epochs used by our method. To make training and inference practical on our hardware, we vectorized the synonymous masked-token loss, and batched independent decoding windows.

Table 9: Within-tree Comparison on a Shared 10,000-Target Cohort

Model	Codon match (%) ↑	$\lvert\Delta\mathrm{GC}\rvert$ (pp) ↓	$\lvert\Delta\mathrm{CFD}\rvert$ (pp) ↓	$\lvert\Delta\mathrm{CSI}\rvert$ ↓	Normalized DTW ↓	$\lvert\Delta\mathrm{MFE}\rvert$ (kcal/mol) ↓
PINATA (Ours)	**65.67 ± 0.02**	**1.453 ± 0.014**	**0.951 ± 0.004**	**0.0216 ± 0.0002**	**4.608 ± 0.020**	**21.30 ± 0.09**
CodonTransformer	50.04 ± 0.45	6.678 ± 0.598	3.146 ± 0.087	0.1100 ± 0.0006	12.306 ± 0.353	86.81 ± 6.17
SynCodonLM	42.78 ± 1.04	11.153 ± 1.431	4.073 ± 0.383	0.1247 ± 0.0131	18.167 ± 2.335	122.97 ± 21.78
PINATA (no MSA ablation)	50.58 ± 0.19	2.610 ± 0.025	1.388 ± 0.009	0.0380 ± 0.0003	5.952 ± 0.029	35.07 ± 1.46
MSA signal only (no learning)	58.41 ± 0.06	3.742 ± 0.011	2.032 ± 0.070	0.0482 ± 0.0008	6.334 ± 0.019	37.18 ± 0.40