DISTILLING GENOMIC MODELS FOR EFFICIENT mRNA REPRESENTATION LEARNING VIA EMBEDDING MATCHING

Rasched Haidari, Sam Martin & Maxime Allard*
Helical
London, UK
{maxime}@helical-ai.com
*Corresponding Author

ABSTRACT

Large Genomic Foundation Models have recently achieved remarkable results and in-vivo translation capabilities. However these models quickly grow to over a few Billion of parameters and are expensive to run when compute is limited. To overcome this challenge, we present a distillation framework for transferring mRNA representations from a state of the art genomic foundation model into a much smaller model specialized for mRNA sequences, reducing the size by 200-fold. Embedding-level distillation worked better than logit based methods, which we found unstable. Benchmarking on mRNA-bench demonstrates that the distilled model achieves state-of-the-art performance among models of comparable size and competes with larger architectures for mRNA-related tasks. Our results highlight embedding-based distillation of mRNA sequences as an effective training strategy for biological foundation models. This enables similar efficient and scalable sequence modelling in genomics, particularly when large models are computationally challenging or infeasible.

1 INTRODUCTION

Distillation is a widely used approach for compressing large models into smaller efficient counterparts while preserving most of their predictive power Hinton et al. (2015); Romero et al. (2015); Zagoruyko & Komodakis (2017); Yim et al. (2017); Tung & Mori (2019); Heo et al. (2019). As model sizes continue to grow, both training and inference become increasingly expensive and computationally demanding. Larger models also suffer from slower inference times, which is problematic in domains requiring large-scale forward passes. For instance, in-silico cell perturbation studies often involve evaluating thousands or potentially millions of computational experiments Lotfollahi et al. (2019); Lopez et al. (2018); Cui et al. (2024). Compact yet capable models are therefore essential for feasibility and cost-effectiveness. Such models can also serve as efficient proxies for rapid verification or filtering before resorting to billion-parameter architectures.

In this paper, we use distillation to train a student model called HelixNano-mRNA (Hybrid Mamba-Attention model Wood et al. (2025) with $\sim$5M parameters) using representations of mRNA sequences from a teacher model, which we chose to be Evo2-1B Brixi et al. (2025). Specifically, we align intermediate embeddings from Evo2 with two student hidden layers to transfer structural and contextual information. While the teacher model is trained on both DNA and RNA sequences, we distill knowledge only on mRNA inputs to obtain a student model specialized for mRNA representation learning. Direct logit distillation Hinton et al. (2015) was found to be unstable, with the KL divergence oscillating heavily during training Yuan et al. (2021). This may be due to our work involving biological data as opposed to text Jiao et al. (2020). Embedding distillation produced consistent improvements yielding state-of-the-art performance among models of comparable size and out-performing a host of much larger models Sun et al. (2019); Jiao et al. (2020).

Little work exists in the post-training landscape for genomic models, and even less so for distillation of these models Madani et al. (2020); Geffen et al. (2022). In this work we show the following contributions:

- Embedding-based distillation, at least for mRNA sequences, is more stable than logit-based distillation for genomic models.
- 200-Fold reduction in size while achieving state-of-the-art performance for its size on `mRNA-bench` Shi et al. (2025).
- Using intermediate latent representations is an effective approach for biological model distillation.

After the distillation, we name the resulting student model HelixNano-mRNA. It can be easily extended with linear layers and a task-specific loss for downstream mRNA sequence generation, classification, and regression. We release the distilled model weights at `https://huggingface.co/helical-ai/HelixNano-mRNA`.

2 METHODS

2.1 DISTILLATION VIA INTERMEDIARY LATENT EMBEDDINGS

In general, the distillation loss function can be expressed as a combination of $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{ED}}$, where $\mathcal{L}_{\text{KL}} = \text{KL}\big(p_t(x) \,\|\, p_s(x)\big)$ is the KL divergence between the output probability distributions of the teacher $p_t(x)$ and student $p_s(x)$ Hinton et al. (2015); Gou et al. (2021). $\mathcal{L}_{\text{ED}}$ is the loss between teacher $E_t(x)$ and student embeddings $E_s(x)$. In our case specifically, we define $\mathcal{L}_{\text{ED}}$ as a combination of both the mean-square loss $\mathcal{L}_{\text{mse}} = \|\|E_t(x) - E_s(x)\|\|_2^2$ and cosine loss $\mathcal{L}_{\cos} = 1 - E_t(x) \cdot E_s(x)$. Empirically in our experiments, removing the KL and cross-entropy terms performed best which is why in the following we only use the embedding matching loss.

Our adapted distillation loss is given by $\mathcal{L}_{\text{train}} = \lambda_{\cos}\,\mathcal{L}_{\cos} + \lambda_{\text{mse}}\,\mathcal{L}_{\text{mse}} + \lambda_{\text{wd}}\,\|\theta\|_2^2$, where $\mathcal{L}_{\cos}$ is the cosine embedding loss between up-projected student embeddings and teacher embeddings and θ denotes all trainable student parameters. We set $\lambda_{\cos} \gg \lambda_{\text{mse}}$ to ensure directional alignment dominates while the Euclidean term prevents exploding norms. Alongside optimizer weight decay this led to better training stability and smaller absolute embedding values.

Focusing on mRNA sequences, we first tokenize mRNA sequences similarly to the original Evo2-1B model (per nucleotide) and feed them to both models, freezing the teachers weights.

To match the different sizes between the teacher and student model we use linear projection layers to align the hidden dimensions of latent teacher and student embeddings (see Figure 1, $\mathbb{R}^{256} \rightarrow \mathbb{R}^{1942}$) Tian et al. (2022); Zbontar et al. (2021); Bardes et al. (2022); Chen et al. (2023); Miles & Mikolajczyk (2024). Down-projection of the teacher embeddings can lead to zero loss solutions as both the model and projection layers collapse. We did not choose layers too early in Evo2 as they may not encode as much latent information and found two layer matching to work better than one layer Sun et al. (2019). Prior work on Genomics Foundation Models has also shown that the best embeddings are not typically from final layers Dalla-Torre et al. (2025); Brixi et al. (2025). For student embeddings we concatenate the output from both layers and the mean over the context length is taken for a one-dimensional final embedding.

As we match final layers of both models, the student may have learnt sufficient representations from Evo2 for mapping back to the token space (e.g., using an additional linear layer). Due to computational limits, we were not able to evaluate how all combinations of layer pairs affect evaluation metrics. Nevertheless, we achieve state-of-the-art performance on a current benchmark using reasonable chosen blocks Shi et al. (2025). We found that taking the embeddings post-norm worked better due to normalisation. We do not expect our model to learn all Evo2 latent representations due to the dimension mismatch but rather learn to encode a number of essential directions.

2.2 DISTILLATION DETAILS

We train with the following hyperparameters: batch size 32, sequence lengths of 2048 tokens, AdamW with a weight decay of 1×10^{-2}, learning rate of 2×10^{-4} with linear warmup over

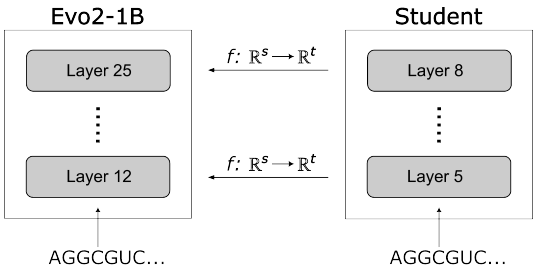


Figure 1: The student model is aligned with two hidden layers (5th and 8th) using projections from layers of Evo2-1B (12th and 25th) .

the first 2000 steps, $\lambda_{\text{cos}} = 1.0$, $\lambda_{\text{mse}} = 0.1$, dropout 10%, and gradient norm clipping with a maximum of 1.0. We train in mixed precision (bfloat16) on 4x A100 GPUs. Total training time is 24 hours.

2.3 Dataset

Training mRNA datasets can be found on the NCBI ftp server: `https://ftp.ncbi.nih.gov/refseq/release/`. We used all the files ending in '.rna.gbff.gz' and subsampled 27 million sequences out of the possible 56 million due to budgetary constraints. The dataset consisted of 43.6% other vertebrates (`https://ftp.ncbi.nih.gov/refseq/release/vertebrate_other/`), 28.3% mammals (`https://ftp.ncbi.nih.gov/refseq/release/vertebrate_mammalian/`), 26.4% invertebrates (`https://ftp.ncbi.nih.gov/refseq/release/invertebrate/`) and 1.6% viruses (`https://ftp.ncbi.nih.gov/refseq/release/viral/`).

2.4 Evaluation and Metrics

During training and evaluation we monitor the below metrics (alongside total loss). Note we use an up-arrow (↑) for student embeddings **after** the linear projection layer.

To assess the effect of the projection layer, we applied the above Centred Kernel Alignment (CKA) metric, which quantifies the structural similarity between two sets of representations (independent of embedding dimensionality Kornblith et al. (2019).

In total we use the following metrics: **Projection cosine loss and MSE** for each aligned layer $\mathcal{L}_{\text{cos}} = 1 - E_s^{\uparrow}(x) \cdot E_t(x)$ and $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(E_s^{\uparrow}(x_i) - E_t(x_i)\right)^2$; **Student embedding variance** to detect collapse: $\text{Var}(E_s) = \frac{1}{d}\sum_{j=1}^{d}\text{Var}(E_{s,j})$; **Linear CKA** between teacher embeddings and (up-projected) student embeddings $\text{CKA}(E_t,\, E_s^{(\uparrow)}) = \frac{\left\|E_t^\top E_s^{(\uparrow)}\right\|_F^2}{\left\|E_t^\top E_t\right\|_F \left\|(E_s^{(\uparrow)})^\top E_s^{(\uparrow)}\right\|_F}$.

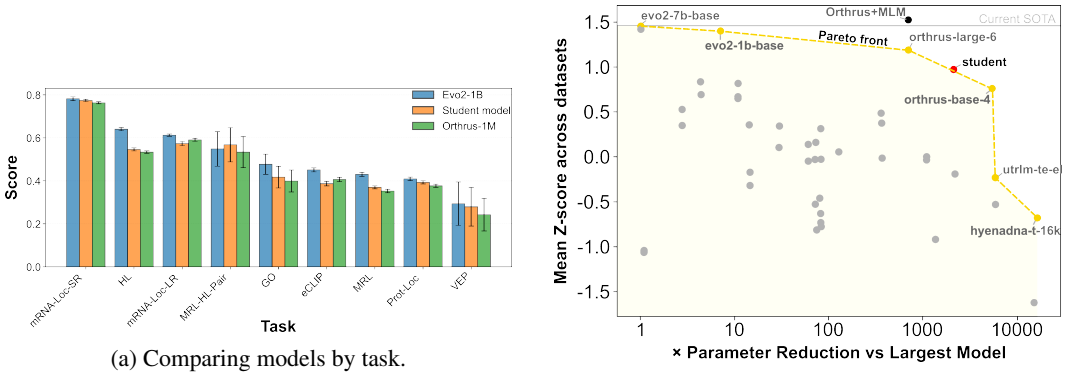
After training, we benchmark the model on `mRNA-bench` Shi et al. (2025), which involves : mRNA Half-Life (**HL**) (measures transcript stability via half-life), mRNA Subcellular Localization (**mRNA-Loc-SR**) (predicts cellular compartment from short-read data; long-read data unavailable [1]), Paired mRNA Half-Life and Mean Ribosome Load (**MRL-HL-Pair**) (jointly models stability and translation efficiency), GO Term Classification (**GO**) (predicts gene function across molecular, biological, and cellular categories), Protein Localization (**Prot-Loc**) (predicts protein subcellular localization) Massively Parallel Translation Assay – Mean Ribosome Load (**MRL-MPRA**) (predicts translation efficiency from synthetic 5'UTRs), eCLIP Binding (**eCLIP**) (predicts RNA-binding protein interactions), and Variant Effect Prediction (**VEP**) (predicts pathogenic single-nucleotide variants in mRNA).

[1] `https://github.com/morrislab/mRNABench/issues/23`

3 RESULTS

We train until convergence with training and validation curves shown on Figure A1. Both losses decrease sharply during the initial phase and then reduce much slower. Similar behaviour is reflected in other metrics (see Appendix). During training gradient norms and variances exploded using only a cosine loss, which is why a small MSE loss was added between student and teacher embeddings (see Figures A2–A4). We computed CKA both before and after projection to evaluate whether the linear layer contributes substantially. Figure A5 shows the CKA values pre- and post-projection, revealing no significant impact from the linear layer.

We benchmarked Evo2-1B, the student model - HelixNano-mRNA, and orthrus-base-4 (Orthrus-1M) Fradkin et al. (2024) on `mRNA-bench`. Score by task are shown in Figure 2a while overall score by model size is shown on Figure 2b. The distilled model achieves leading performance across nearly all tasks for its size [2].



(a) Comparing models by task.

(b) Overall model performance by size. Our model is denoted by red dot. Figure adapted and re-created from Shi et al. (2025).

Figure 2: Benchmark results.

Distillation was attempted using sequence logits and KL divergences but this did not perform as well. The entropy for the logit distribution was very noisy, including large spikes and erratic behaviour, making model learning difficult particularly given its smaller size (see Figure A6 and Figure A7). We briefly tested the model by matching a single hidden layer with Evo2 but found better results with two layer matching. This is not an extensive analysis and more work should be done to verify this behaviour Yu et al. (2025).

For the matched layers we implemented PCA to the Evo2 embeddings giving us the number of dimensions needed to account for $> 90\%$ of the variance. This was a surprisingly small for post-norm layers (6 dimensions) but much higher for block 12 (431, see Table 1) which was used for downstream biological tasks by the Evo2 authors Brixi et al. (2025). The norm layers are most likely collapsing the embedding space, explaining why the student model was able to achieve lower cosine losses as compared to non-norm layers. A UMAP of embeddings by phylum and model is shown in the Appendix (see Figure A8).

4 FUTURE WORK

This work presents HelixNano-mRNA and aims to set up foundations for distillation and other post-training work in the biological foundational model domain. Future work can focus on both further interpreting noisy KL divergences in distillation of biological sequences alongside effects of matching various layers. Extending the distilled embeddings to downstream tasks, such as sequence generation and variant effect prediction will further evaluate predictive performance.

[2] mRNA-Loc-LR omitted from aggregated results due to unavailability of the processed version used in Shi et al. (2025) at time of writing; see `https://github.com/morrislab/mRNABench/issues/23`

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022. URL https://arxiv.org/abs/2105.04906.

Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, Mohsen Naghipourfar, Eric Nguyen, Chiara Ricci-Tam, David W Romero, Gwanggyu Sun, Ali Taghibakshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K Wang, Etowah Adams, Stephen A Baccus, Steven Dillmann, Stefano Ermon, Daniel Guo, Rajesh Ilango, Ken Janik, Amy X Lu, Reshma Mehta, Mohammad R.K. Mofrad, Madelena Y Ng, Jaspreet Pannu, Christopher Re, Jonathan C Schmok, John St. John, Jeremy Sullivan, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Tom McGrath, Kimberly Powell, Dave P. Burke, Hani Goodarzi, Patrick D Hsu, and Brian Hie. Genome modeling and design across all domains of life with evo 2. *bioRxiv*, 2025. doi: 10.1101/2025.02.18.638918. URL https://www.biorxiv.org/content/early/2025/02/21/2025.02.18.638918.

Yudong Chen, Sen Wang, Jiajun Liu, Xuwei Xu, Frank de Hoog, and Zi Huang. Improved feature distillation via projector ensemble, 2023. URL https://arxiv.org/abs/2210.15274.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 2024. doi: 10.1038/s41592-024-02201-0.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z. Epub 2024 Nov 28.

Philip Fradkin, Ruian Shi, Keren Isaev, Brendan J. Frey, Quaid Morris, Leo J. Lee, and Bo Wang. Orthrus: Towards evolutionary and functional rna foundation models. *bioRxiv*, 2024. doi: 10.1101/2024.10.10.617658.

Yaron Geffen, Yanay Ofran, and Ron Unger. Distilprotbert: a distilled protein language model used to distinguish between real proteins and their randomly shuffled counterparts, 2022.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation, 2019. URL https://arxiv.org/abs/1904.01866.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020. URL https://arxiv.org/abs/1909.10351.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019. URL https://arxiv.org/abs/1905.00414.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Mohammad Lotfollahi, Fabian A Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019. doi: 10.1038/s41592-019-0494-8.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation, 2020. URL https://arxiv.org/abs/2004.03497.

Roy Miles and Krystian Mikolajczyk. Understanding the role of the projector in knowledge distillation, 2024. URL https://arxiv.org/abs/2303.11098.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015. URL https://arxiv.org/abs/1412.6550.

Ruian Shi, Taykhoom Dalal, Philip Fradkin, Divya Koyyalagunta, Simran Chhabria, Andrew Jung, Cyrus Tam, Defne Ceyhan, Jessica Lin, Kaitlin U. Laverty, Ilyes Baali, Bo Wang, and Quaid Morris. mrnabench: A curated benchmark for mature mrna property and function prediction. *bioRxiv*, 2025. doi: 10.1101/2025.07.05.662870. URL https://www.biorxiv.org/content/early/2025/07/08/2025.07.05.662870.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression, 2019. URL https://arxiv.org/abs/1908.09355.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2022. URL https://arxiv.org/abs/1910.10699.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation, 2019. URL https://arxiv.org/abs/1907.09682.

Matthew Wood, Mathieu Klop, and Maxime Allard. Helix-mrna: A hybrid foundation model for full sequence mrna therapeutics, 2025. URL https://arxiv.org/abs/2502.13785.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Zony Yu, Yuqiao Wen, and Lili Mou. Revisiting intermediate-layer matching in knowledge distillation: Layer-selection strategy doesn't matter (much), 2025. URL https://arxiv.org/abs/2502.04499.

Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization, 2021. URL https://arxiv.org/abs/1909.11723.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2017. URL https://arxiv.org/abs/1612.03928.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. URL https://arxiv.org/abs/2103.03230.

A APPENDIX

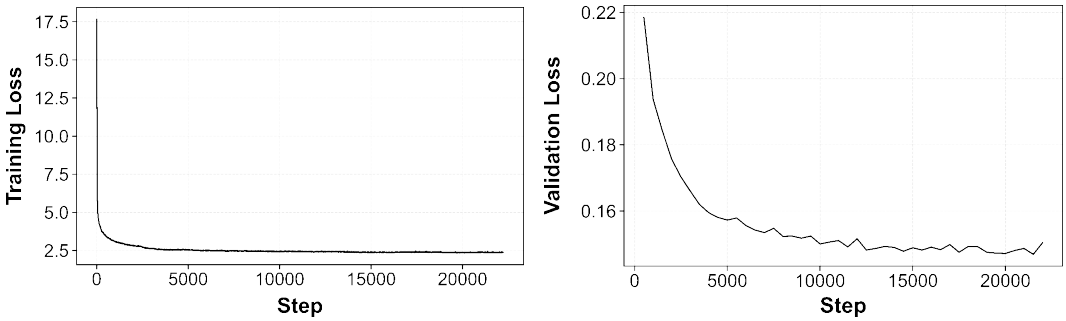


Figure A1: Losses for the train and validation set

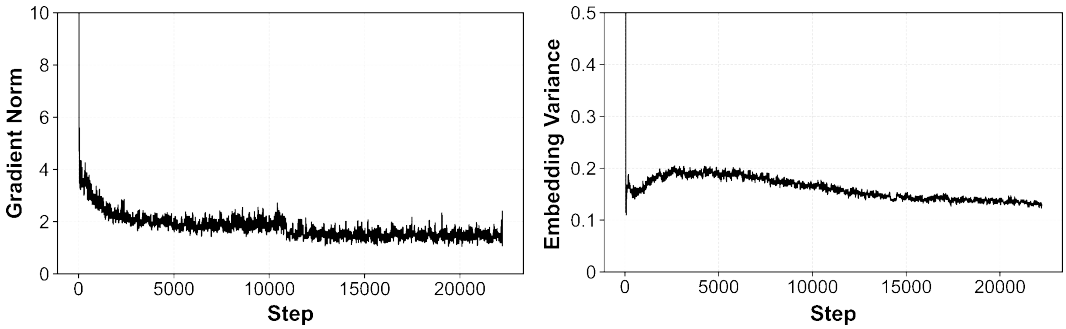


Figure A2: Student model gradient norms and embedding variances during training

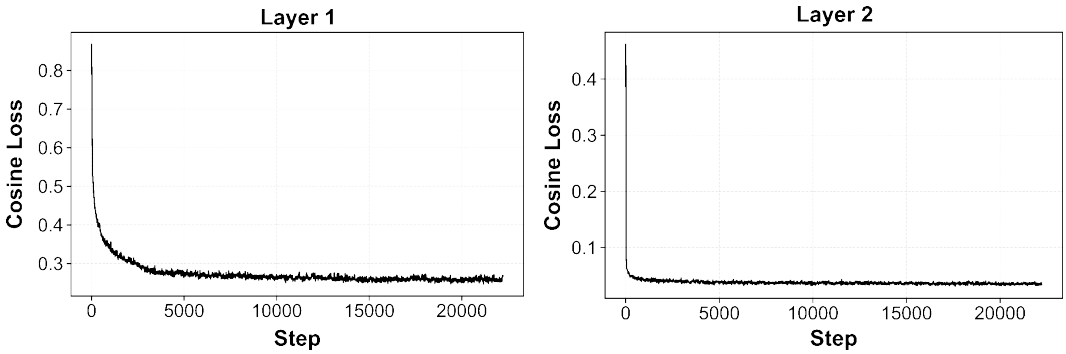


Figure A3: Cosine training loss for the first and second matched layers.

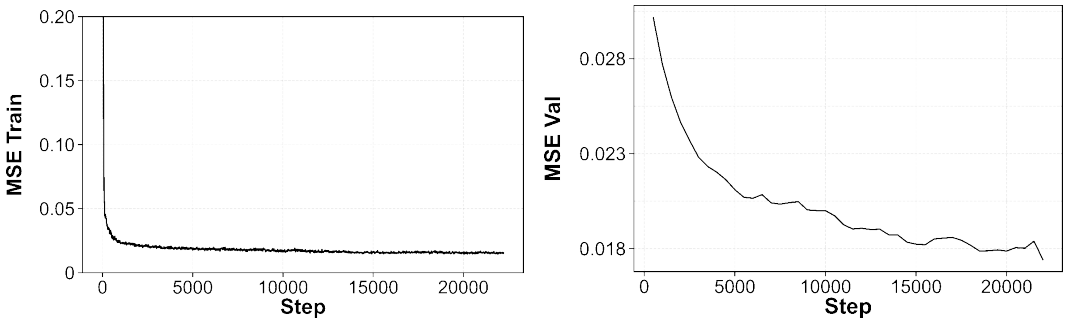


Figure A4: MSE Losses for training and validation data

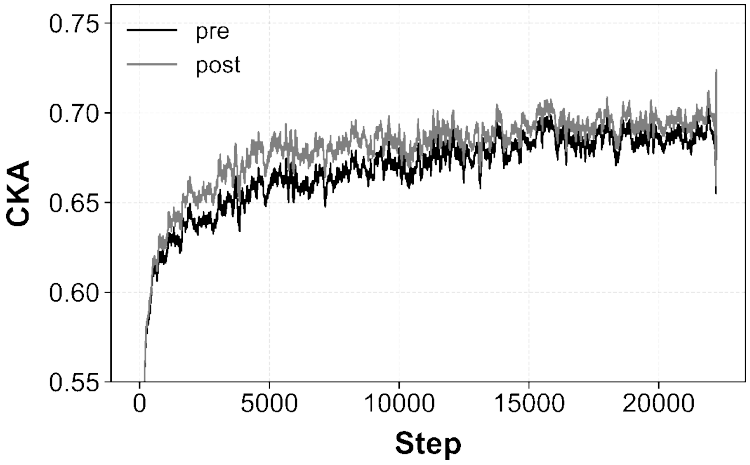


Figure A5: Central Kernel Alignment (CKA) Values Pre- and Post- Linear Projection Layer during training.

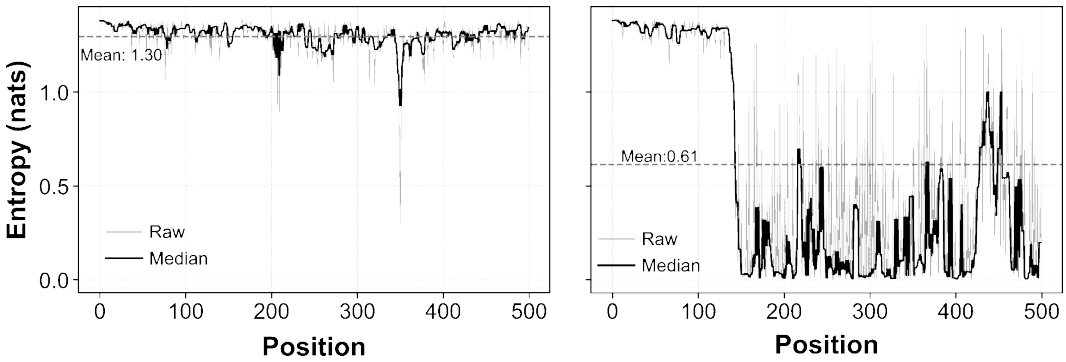


Figure A6: Entropy against position from the start of the mRNA sequence. Abrupt changes in the logit distribution and regions of high entropy make learning difficult.

Table 1: Number of principal components required to reach different variance thresholds for Evo2 hidden layers.

Variance Threshold	block 12	norm
50%	1	2
75%	1	4
90%	1	5
95%	1	6
99%	431	6

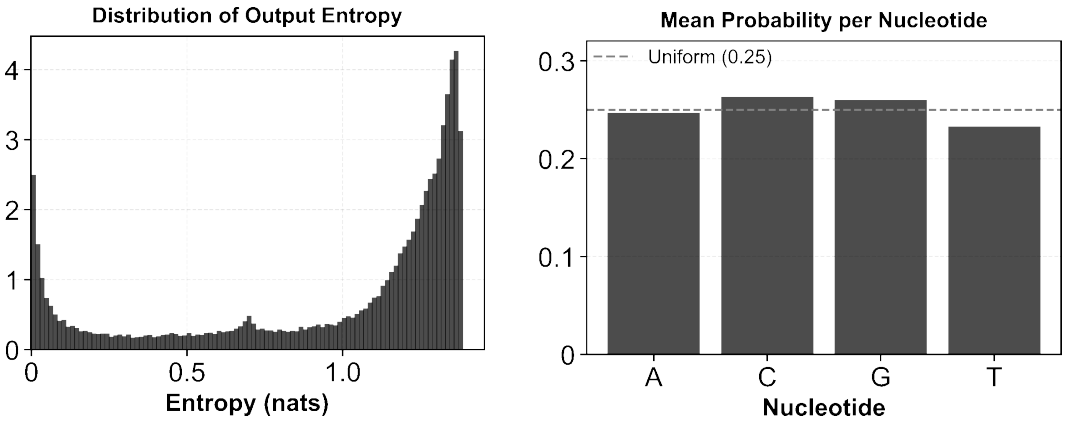


Figure A7: Summary entropy statistics across 100 mRNA sequences. The average probability per token is slightly better than uniform.

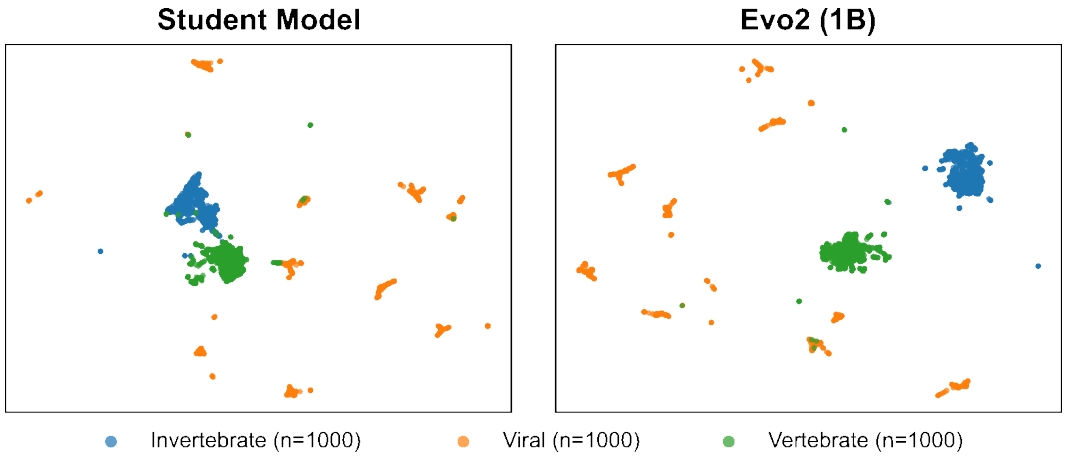


Figure A8: UMAP over Phylum for Evo2 and the student model.