BIRDIEDNA: REWARD-BASED PRE-TRAINING FOR GENOMIC SEQUENCE MODELING

Sam Blouir[1] **Defne Circi**[2] **Asher Moldwin**[1] **Amarda Shehu**[1]

[1]George Mason University, Fairfax, VA, USA
[2]Duke University, Durham, NC, USA

sblouir@gmu.edu, defne.circi@duke.edu

ABSTRACT

Transformer-based language models have shown promise in genomics but face challenges unique to DNA, such as sequence lengths spanning hundreds of millions of base pairs and subtle long-range dependencies. Although next-token prediction remains the predominant pre-training objective (inherited from NLP), recent research suggests that multi-objective frameworks can better capture complex structure. In this work, we explore whether the Birdie framework, a reinforcement learning-based, mixture-of-objectives pre-training strategy, can similarly benefit genomic foundation models. We compare a slightly modified Birdie approach with a purely autoregressive, next token prediction baseline on standard Nucleotide Transformer benchmarks. Our results show performance gains in the DNA domain, indicating that mixture-of-objectives training could be a promising alternative to next token prediction only pre-training for genomic sequence modeling.

1 INTRODUCTION

Genomic sequences encode the fundamental blueprint of life, guiding complex regulatory processes and influencing phenotypic traits. Unlike most natural language processing (NLP) tasks that generally span from a few hundred to several hundred thousands of tokens, genomic sequences can extend over hundreds of millions of nucleotides, each of which are often tokenized alone (Poli et al., 2023). These immense lengths, combined with the subtlety of genomic signals, pose significant computational and representational challenges.

Transformer-based architectures have pushed the boundaries of sequence modeling in NLP. Unfortunately, their quadratic complexity with regards to the sequence length can hinder direct application to genomic data. Recent works (Nguyen et al., 2024; 2023; Poli et al., 2023) propose novel state-space and kernel-based models or hybrid architectures that can handle longer sequences efficiently. However, beyond particular architecture innovations, *training objectives* themselves can significantly influence the learned representations of the model.

Recently, the Birdie framework (Blouir et al., 2024) introduced a **mixture-of-objectives** pre-training strategy driven by reinforcement learning, combining classic and new objectives, including infilling and prefix language modeling Raffel et al. (2020); Tay et al. (2023b). A reward model was used to learn associations between objective sampling ratios and per-objective delta losses after a period of training steps. Random per-objective sampling ratios, called actions, were then fed to the reward model to estimate which sampling ratios should be used for the next amount of training steps. This approach significantly improved long-range retrieval and text comprehension for NLP models on tasks like SQuAD-v2, story causality comprehension, and multi-phone number retrieval.

In this work, we adopt and slightly adapt the Birdie framework for genomic sequences. Our experiments show that mixing multiple objectives consistently outperforms single-objective next-token prediction baselines, especially under data-scarce conditions.

The key contributions from our paper are:

- Adapting the Birdie mixture-of-objectives framework for the genomic domain.

- Evaluating performance on standard Nucleotide Transformer tasks, demonstrating improvement over the Next Token Prediction baseline.
- A discussion of challenges when translating methods between the NLP and Genome domains.

2 Related Work

2.1 Genomic Sequence Modeling

Early methods in genomics often relied on handcrafted features or simpler neural networks limited to short sequences. Recent deep learning efforts transitioned to CNNs and RNNs (Alipanahi et al., 2015; Zhou and Troyanskaya, 2015), then to Transformers (Ji et al., 2021), and more recently to advanced state-space and kernel-based architectures (Gu et al., 2021; Nguyen et al., 2023). Despite architectural innovation, long context length and relevant, diverse tasks and objectives for pre-training continue to be obstacles for large genome models.

2.2 Mixture-of-Objectives Training

Training on Next Token Prediction can lead to reduced performance on downstream tasks compared to training with a diverse mixture of objectives Tay et al. (2023a); Lewis et al. (2019); Blouir et al. (2024). Training with a mixture of objectives can help the model learn abstract versions of relevant skills and prime them for downstream tasks. This can be particularly useful for situations where a downstream task does not have many training samples.

3 Methodology

3.1 Birdie Mixture-of-Objectives

The Birdie framework trains models on multiple training objectives and tasks simultaneously (Blouir et al., 2024). Originally designed for NLP tasks, we adjust and simplify the objectives for our DNA sequences, following several pilot runs. First, we remove Next Token Prediction entirely. We find that our small Transformers suffered from reducced overall performance with the presence of this objective. Second, we use two aggressive infilling configurations from UL2R used in U-PaLM (Tay et al., 2022), specifically using only 50% token corruption with an average span width of 3 tokens, or, an average of 15% token corruption with an average span width of 32 tokens. We keep all other objectives in the original Birdie framework. We describe all objectives below:

- **Infilling:** Randomly mask spans of the input and train the model to generate them. This task can help the model capture context-sensitive dependencies from across the sequence, as well as handle partially corrupted input data. We use the aforementioned settings: an average span width of 3 tokens with 50% of input tokens masked out, or we mask out 15% of the tokens with an average span width of 32.
- **Next Token Prediction:** Generate the next token from left to right, in a causal manner. This approach is the standard autoregressive language modeling and helps the model learn sequential dependencies.
- **Deshuffling:** Randomly shuffle the order of subsequences within the input, and train the model to restore the original sequence. This objective ensures the model recognizes structural coherence and ordering. We shuffle 50% of the input sequence.
- **Prefix Language Modeling:** Provide a partial prefix of the input and train the model to predict the remaining content. By learning to continue truncated sequences, the model handles arbitrary contexts and completes them coherently. Following UL2 and Birdie, we set the bidirectioal prefix area to be 75% of the input sequence. Therefore, the model generates the last 25% of the sequence with no loss calculated on the prefix.
- **Autoencoding:** This objective is similar to infilling, where spans of tokens have been replaced with a mask token Lewis et al. (2019). We reconstruct the entire input. This entails copying the unmasked spans from the original sequence. We replace 30% of the

input tokens with several spans. The model re-produces the original input sequence. We also shuffle an average of 50% of the non-masked input spans, creating a very similar setup to the best objective found in BART Lewis et al. (2019).

- **Copying:** Generate the exact input. Although simple, this objective ensures the model is capable of producing lossless reconstructions, serving as a baseline or supplementary task.
- **Selective Copying:** The model is given the prefix and suffix of a string to retrieve from the context. This trains the model to be able to locate and selectively copy spans of text. We use 15% of the input text as spans. This can be seen as a structured shuffling of unshuffled input data.

3.2 Pretraining

We pre-train two four layer Transformers. We place further model details and specifics in appendix subsection 8.1.

3.2.1 Training Data: T2T-CHM13

We pre-train on the T2T-CHM13 human genome assembly (Nurk et al., 2022), which is roughly 3.1 billion nucleotides in length. This is one of the newest versions of the human genome.

3.3 Evaluation Datasets and Tasks

We use the latest Nucleotide Transformer benchmarks (Dalla-Torre et al., 2023). We provide descriptions in section 8.

4 Experiments

4.1 Setup and Baseline

We compare:

- **NTP-Only Baseline:** A model trained *exclusively* on next-token prediction.
- **BirdieDNA (Mixture-of-Objectives):** Our approach sampling masked, infilling, prefix, deshuffling, selective copying, and other tasks from Birdie. We have slightly modified them, as described in subsection 3.1.

Each downstream task was trained on directly for 8,192 steps with a batch size of 32. We used a fixed learning rate of 5e-5 with no weight decay. We collect the evaluation performance every 256 steps.

4.2 Results

We find that the Birdie-trained model reached higher accuracies significantly faster than the Next Token Prediction-trained model. Our experimental setup used a total of 262,144 samples per run, from 8,192 finetuning steps and a batch size of 32. The majority of these tasks have 30,000 training samples, so this represents at least 8.7 epochs. These F1 and Accuracies may be evidence that Birdie is less necessary should the downstream task allow for a specialized set of model weights, These results diverge from the original Birdie paper, where certain models were ”stuck” in local minima following Blouir et al. (2024)

Following similar work Poli et al. (2023); Dalla-Torre et al. (2023), we do not report overall averages. We report F1 and Accuracy for each benchmark from the latest version of the Nucleotide Transformer's benchmark collection Dalla-Torre et al. (2023).

4.3 Discussion

Our findings show a slight, but present, improvement on model performance from our mixture-based training. Pre-training on self-supervised, abstract versions of these tasks can show up as improved

Task Name	Model	F1 (%)	Accuracy (%)
Promoter All	Birdie	83.31	**83.93**
	NTP	**83.38**	83.79
Promoter No Tata	Birdie	**84.74**	**84.85**
	NTP	84.63	84.77
Promoter Tata	Birdie	**92.44**	**91.67**
	NTP	87.80	86.11
Splice Sites Donors	Birdie	84.75	83.38
	NTP	**86.11**	**84.52**
Enhancers	Birdie	70.40	68.07
	NTP	**71.02**	**68.14**
H2Afz	Birdie	**72.03**	**67.67**
	NTP	71.97	67.37
H3K27Ac	Birdie	**70.58**	**65.75**
	NTP	70.20	65.25
H3K27Me3	Birdie	75.47	**71.90**
	NTP	**75.51**	71.80
H3K36Me3	Birdie	77.00	73.20
	NTP	**77.43**	**73.60**
H3K4Me1	Birdie	**71.29**	**65.97**
	NTP	71.21	65.63
H3K4Me2	Birdie	72.83	**69.58**
	NTP	**73.53**	69.50
H3K4Me3	Birdie	75.36	75.75
	NTP	**75.95**	**76.25**
H3K9Ac	Birdie	72.69	68.25
	NTP	**73.77**	**69.50**
H3K9Me3	Birdie	**68.00**	57.75
	NTP	65.20	**58.75**

Table 1: **Performance metrics for each task (F1 and Accuracy).** We compare our Birdie-trained Transformer with a Next Token Prediction-only trained Transformer. These results come from the latest version of the Nucleotide Transformer benchmarks Dalla-Torre et al. (2023); Poli et al. (2023) and are not directly comparable to older literature.

performance on downstream tasks, and these results align with previous literature (Tay et al., 2023b; Raffel et al., 2020; Blouir et al., 2024) showing multi-task or mixture-based pre-training can result in more robust and flexible models.

5 CONCLUSION

We present BirdieDNA, a mixture-of-objectives training strategy for genomic sequence modeling. By sampling multiple reconstruction and prediction tasks, our method produces robust representations that transfer well to key downstream genomics tasks. Experimental results suggest consistent improvements across enhancer detection, promoter classification, splice-site prediction, histone mark classification, and variant effect prediction. In future work, we envision integrating additional objectives (e.g., evolutionary or structural constraints) and exploring larger-scale multi-species training corpora. Mixture-of-objectives training represents a straightforward yet powerful route to tackling the immense complexity of genomic sequences.

6 LIMITATIONS

The results are relatively speaking, close. We believe is a limitation of the datasets and procedures for these common DNA downstream tasks, as compared to NLP. After we ablated many hyperparameters to get the best result for each pre-training framework, the settings that led to the best results for each task may not have been compatible with simply training on collections of these tasks, as is common with transfer learning in NLP Longpre et al. (2023).

7 FAILURES

We also attempt to perform a second-stage pretraining of Llama 3.2 1B to handle DNA sequences. Whether or not we used a mixture-of-denoisers approach or standard next-token prediction objective, the model was unable to out-perform a smaller freshly initialized model on these tasks. We hypothesize that this may be due to an excessive model size for the size and ”grokkability” of the datasets used.

REFERENCES

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* 33, 8 (2015), 831–838.

Sam Blouir, Jimmy TH Smith, Antonios Anastasopoulos, and Amarda Shehu. 2024. Birdie: Advancing State Space Models with Reward-Driven Objectives and Curricula. *arXiv preprint arXiv:2411.01030* (2024).

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. 2023. The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics. *bioRxiv* (2023). https://doi.org/10.1101/2023.01.11.523679 arXiv:https://www.biorxiv.org/content/early/2023/01/15/2023.01.11.523679.full.pdf

Albert Gu and Tri Dao. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs.LG]

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396* (2021).

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 37, 15 (2021), 2112–2120.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461 [cs.CL] https://arxiv.org/abs/1910.13461

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arXiv:2301.13688 [cs.AI]

Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. 2024. Sequence modeling and design from molecular to genome scale with Evo. *Science* 386, 6723 (2024), eado9336.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. 2023. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems* 36 (2023), 43177–43201.

Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V Bzikadze, Alla Mikheenko, Mitchell R Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, et al. 2022. The complete sequence of a human genome. *Science* 376, 6588 (2022), 44–53.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, et al. 2023. Hyena Hierarchy: Towards Larger Convolutional Language Models. arXiv:2302.10866 [cs.LG]

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG]

Noam Shazeer. 2020. GLU Variants Improve Transformer. arXiv:2002.05202 [cs.LG]

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Tran, Dani Yogatama, and Donald Metzler. 2023a. Scaling Laws vs Model Architectures: How does Inductive Bias Influence Scaling?. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12342–12364. https://doi.org/10.18653/v1/2023.findings-emnlp.825

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023b. UL2: Unifying Language Learning Paradigms. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=6ruVLB727MC

Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q Tran, David R So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, et al. 2022. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399* (2022).

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

Jian Zhou and Olga G Troyanskaya. 2015. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods* 12, 10 (2015), 931–934.

8 APPENDIX

demo_coding_vs_intergenomic_seqs: Distinguish coding sequences (i.e., from protein-coding genes) from non-coding intergenic regions. The dataset combines two sources:

- `intergenomic_seqs_50k.csv` – 50,000 sequences (200 bp) from non-coding regions.
- `random_transcripts.csv` – 50,000 sequences (200 bp) from coding regions.

demo_human_or_worm: Classify sequences as either human or *C. elegans*, using 50,000 200 bp segments from each organism. The goal is to detect species-specific sequence patterns.

dummy_mouse_enhancers_ensembl: Discriminate between mouse enhancer sequences and random genomic background. Each 200 bp sequence is labeled as an enhancer or non-enhancer.

human_enhancers_cohn: Identify human enhancer regions vs. non-enhancer background sequences. Like the mouse enhancer task, each sequence is 200 bp and labeled for enhancer activity.

human_enhancers_ensembl: Similar to the previous human enhancer dataset but curated from Ensembl annotations. The task again is enhancer identification within human DNA.

human_ensembl_regulatory: Classify human DNA segments into specific regulatory categories, including enhancer regions, open chromatin regions (OCRs), and promoter regions.

human_nontata_promoters: Distinguish promoter sequences (that lack a canonical TATA box) from non-promoter regions. Each input is a 200 bp stretch of DNA annotated for its regulatory function.

human_ocr_ensembl: Determine whether a given 200 bp human sequence maps to an open chromatin region (OCR) or a random genomic background location.

8.1 PRETRAINING AND MODEL CONFIGURATION

We pretrain all models on the T2T dataset for 16,384 steps. We use a batch size of 128, and a cosine LR decay from 1e-3 to 1e-5. We use a hidden size of 512, 8 heads of 64 dimensions each, and rotary position encodings with a decay rate of 500,000, as popularized by Gemma 2 Team et al. (2024). We use a standard Transformer++ setup: pre-norm Attention layers followed by SwiGLU MLPsShazeer (2020); Gu and Dao (2023).