BEYOND TOKENS: READ AND PREDICT N-GRAM SPANS WITH A SHARED DICTIONARY FOR GENOMIC FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models lock in a fixed tokenizer and next token prediction before pretraining, fixing the granularity, boundaries, and horizon of what they read and predict. This is restrictive for DNA, in which the relevant signals are sparse, overlapping, multiscale, and context-dependent. To exploit the compact alphabet and the informative chunks intrinsic to DNA, we introduce **Beyond Tokens**, a coadaptive dictionary of n-grams that enables autoregressive models to read and predict high order patterns of variable length and arbitrary position on top of a base level interface. Rather than redefining the sequence units, we treat each base token as an anchor for two directional operations: contextual pattern reading and future pattern prediction. On the input side, SpanReader retrieves context-relevant n-grams of variable length and offset via sparse Top K routing and adds the routed residual at the backbone input. On the output side, SpanPredictor forecasts multihorizon n-grams as atomic targets through a length-biased confidence gate operating over multiscale chunk losses, providing higher order supervision. The two modules share the same dictionary parameters, allowing the patterns employed for reading and those employed for prediction to coevolve throughout pretraining. We instantiate Beyond Tokens within a 1.2B genomic model and evaluate it through supervised frozen-representation probing, where it improves the matched-budget baseline and attains the highest mean among the compared models, surpassing the reported means of 10B and 40B baselines trained on more tokens. These results indicate that models can adaptively learn higher order tokens beyond a fixed vocabulary, jointly for reading and prediction, a capability naturally suited to small-alphabet domains.

1 INTRODUCTION

Most sequence language models precommit the units presented to the backbone, and autoregressive models usually also precommit the output horizon to the next unit. These choices fix, for every position, the granularity at which context is consumed and, for autoregressive models, the horizon at which predictions are issued. Genomic foundation models, hereafter GFMs, instantiate the input side with single nucleotides, fixed k-mers, or learned subwords (Ji et al., 2021; Zhou et al., 2024; Nguyen et al., 2023; Sanabria et al., 2024); masked encoders recover masked units, whereas conventional autoregressive GFMs predict one base or token at a time.

DNA contains sparse, overlapping functional patterns at multiple lengths and offsets (Avsec et al., 2021; Linder et al., 2025; Avsec et al., 2026; Marin et al., 2024): CpG dinucleotides occur within longer transcription factor motifs, the splice-site cores GT and AG are embedded in wider consensus sequences, and a single base can participate in several functional patterns. A fixed tokenizer chooses sequence units before pretraining, leaving the backbone to construct longer, overlapping patterns from those units (Fig. 1(a)). This motivates explicit multiscale spans both as context representations and as prediction targets. In an evaluation of four pretrained DNA language models across six functional genomics tasks, frozen representations did not substantially outperform one-hot baselines (Tang et al., 2025). Complementary approaches exploit functional supervision (Avsec et al., 2021; Linder et al., 2025) or multispecies alignments (Benegas et al., 2025).

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.

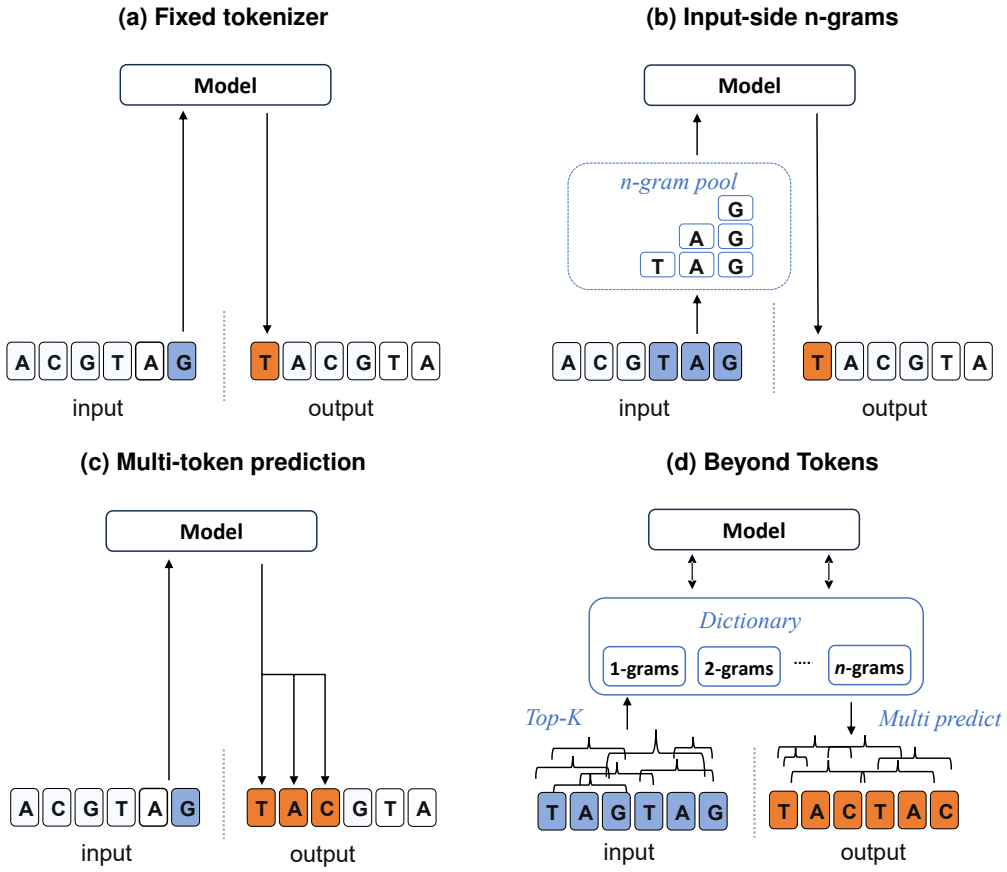


Figure 1: Four paradigms for committing input/output units. **(a)** a fixed tokenizer locks both ports; **(b)** input-side n-gram lookup, with the output remaining at the token level; **(c)** multi-token prediction through multihorizon heads, each predicting a single token; **(d) Beyond Tokens**: a shared n-gram dictionary is consulted via sparse Top K routing at the input port and predicted as atomic chunks at the output port.

The small DNA alphabet makes explicit chunk representations tractable on both sides of the model. With the five-symbol A/C/G/T/N alphabet, a length-6 dictionary has $5^6 = 15{,}625$ entries, whereas a joint n-token target in natural language grows as $|V|^n$; accordingly, prior language-model approaches expand the input vocabulary with multi-grams (Fig. 1(b)) and/or add separate token-level classifiers at several future positions (Fig. 1(c)) (Huang et al., 2025; Gloeckle et al., 2024). In DNA, we can instead supervise atomic higher-order chunks while retaining the base-level autoregressive interface.

We propose **Beyond Tokens** (Fig. 1(d)), which couples context-adaptive span retrieval and atomic future-chunk prediction through the same n-gram embedding tables. The reader selects among causal candidates with different lengths and offsets; the predictor supervises future chunks at multiple horizons. Both operations update one multiscale dictionary, so the representations used to read context are trained jointly with those used as prediction targets.

Our contributions are as follows:

A shared dictionary couples reading and prediction. The same multiscale n-gram parameters serve as context memory and chunk-classifier weights, connecting both learning signals while preserving base-level sequence alignment.

Two complementary modules make this coupling operational. SpanReader selects variable-length, variable-offset spans through sparse routing with reverse-complement-aware keys;

SpanPredictor supplies atomic chunk targets across multiple horizons with length-balanced supervision.

Matched-budget experiments support both modules and their coupling. The full model raises mean AUC from 0.8642 to 0.8806 across 13 tasks; untying the dictionary reduces it to 0.8746. Separate comparisons with reported larger-model results place the full model highest in task mean among the compared models.

2 Related Work

Genomic sequence units Masked GFMs predefine k-mer or BPE units (Ji et al., 2021; Zhou et al., 2024; Sanabria et al., 2024), while long-context autoregressive models use single nucleotides (Nguyen et al., 2023; Brixi et al., 2026) or fixed k-mers (Wu et al., 2025). Learned tokenizers use vector quantization (Li et al., 2024), adaptive convolutions (Qiao et al., 2024), or motif-informed vocabularies (Zhou et al., 2025). Beyond Tokens instead preserves nucleotide alignment and exposes overlapping spans of several lengths, leaving local unit choice to a per-position router. Reverse-complement structure has been handled through parameter sharing (Shrikumar et al., 2017), equivariance (Mallet & Vert, 2021; Schiff et al., 2024), or consistency regularization (Ma, 2025).

Input-side multi-gram memory Over-Tokenized Transformers decouple input and output vocabularies through multi-gram over-encoding and over-decoding, with MTP approximating the latter (Huang et al., 2025). Engram, X-GRAM, and LongCat scale n-gram memories with hashed or frequency-aware lookup schemes (Cheng et al., 2026; Chen et al., 2026; Liu et al., 2026). Their n-gram lookup modules expand read-side capacity. In genomics, DNAZEN dynamically matches corpus-mined G-grams alongside basic units (Mao et al., 2025), while Gengram aggregates multi-scale k-mer lookups (Xu et al., 2026). Beyond Tokens additionally selects spans per position and reuses each learned n-gram table as atomic output class weights.

Future-token prediction and tying Future-token objectives use parallel prediction streams or independent output heads (Qi et al., 2020; Gloeckle et al., 2024), or sequential prediction modules (DeepSeek-AI et al., 2024). These objectives expand prediction horizon rather than output-unit granularity. GENERator-v2 instead derives nucleotide likelihoods from fixed 6-mer logits (Li et al., 2026). Classical tying reuses input parameters in the output layer (Press & Wolf, 2017; Baevski & Auli, 2019), whereas T-FREE shares hashed character-trigram activation patterns but uses separate input-embedding and LM-head parameters (Deiseroth et al., 2024). Beyond Tokens ties multiscale n-gram embeddings across sparse reading and atomic chunk prediction.

3 Method

Fig. 2 sketches Beyond Tokens. We add two ports to a base-level autoregressive backbone, shown as a single block because the modules connect only to its input and hidden state. SpanReader uses sparse Top K routing to inject multiscale motif spans before the first Transformer layer; SpanPredictor applies tied chunk-level heads as auxiliary supervision. Both share one dictionary, with each sequence mapped to an embedding vector: scale one reuses the backbone embeddings for A, C, G, T, and N, while higher scales encode multibase n-grams; other vocabulary tokens are masked. Training combines the language-model, chunk-prediction, and floor-based load-balancing losses, each with a scalar weight.

3.1 SpanReader

At each position, SpanReader forms a query from causal depthwise convolutions at multiple kernel widths, capturing motif context at several length scales without breaking causality. Within a sliding window, each scale-and-offset combination enumerates one motif candidate: a length n subsequence ending at some recent offset relative to the current position. Single base candidates reuse the backbone embedding; multi base candidates retrieve a learnable beyond-token embedding indexed by the forward strand identifier, then project it into the query space.

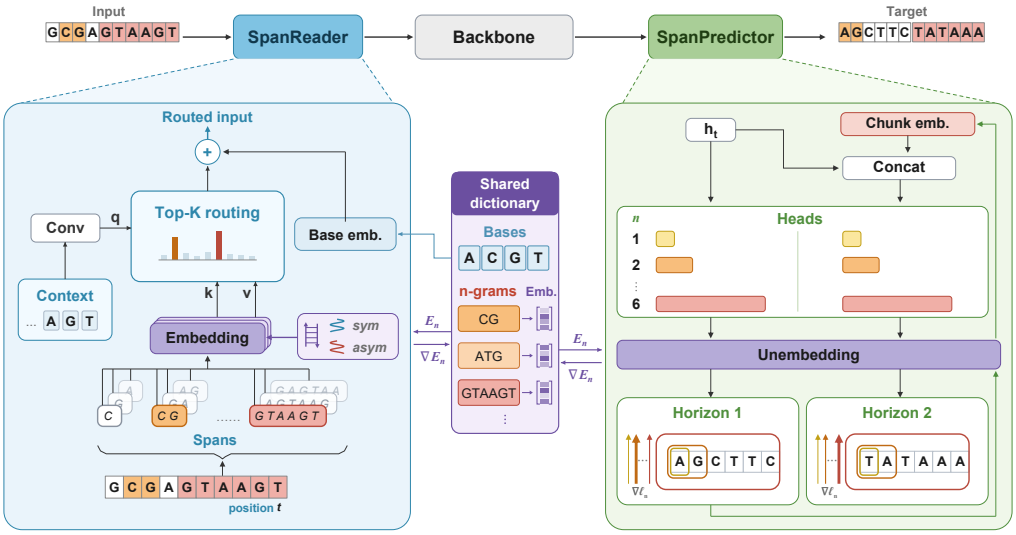


Figure 2: Beyond Tokens architecture. SpanReader (left) injects sparsely routed motif spans; SpanPredictor (right) predicts future chunks using the same dictionary (center). Colors encode span length n and track motifs across modules; gradient-arrow width scales with the length gate g_n.

DNA carries an exact reverse complement symmetry, abbreviated RC below, that token-level models do not respect. In Fig. 2, the paired-strand icon denotes RC-aware key construction: forward and RC embeddings $\mathbf{m}_{fw}, \mathbf{m}_{rc}$ for each multi base candidate yield symmetric and asymmetric components:

$$\begin{aligned} \text{sym} &= \tfrac{1}{2}(\mathbf{m}_{fw} + \mathbf{m}_{rc}), \\ \text{asym} &= \|\mathbf{m}_{fw} - \mathbf{m}_{rc}\|_2 \, \mathbf{D}_n[\text{fw_id}], \\ \mathbf{K}_{n,d} &= \text{sym} + \text{asym}, \end{aligned} \tag{1}$$

with $\mathbf{D}_n$ a learnable per-id direction. Under RC action, sym and the asymmetric magnitude are invariant, so global motif identity is exposed in a strand-agnostic way; only $\mathbf{D}_n[\text{fw_id}]$ remains strand-specific, preserving expressiveness for signals such as splice donor and acceptor sites whose meaning depends on strand. A strict equivariant variant that drops the asymmetric direction is included as an ablation.

The Top-K routing block L2-normalises the query to $\tilde{\mathbf{Q}}$ and scores candidates at a learnable temperature θ_{scale}; the top K scores are softmax-normalised and used to mix value $\mathbf{V}_j = \mathbf{W}_v \mathbf{m}_{n,d}$ into a residual injection added before the first Transformer layer:

$$\begin{aligned} s_j &= \theta_{\text{scale}} \, \tilde{\mathbf{Q}} \cdot \mathbf{K}_j, \\ \mathbf{w} &= \text{softmax}(\text{Top}_K(\mathbf{s})), \\ \mathbf{e}' &= \mathbf{e} + \sum_{j \in \text{Top}_K} w_j \, \mathbf{V}_j. \end{aligned} \tag{2}$$

Keys are deliberately not normalised, so motif magnitude is preserved as part of the routing signal. Top K is preferred over a dense softmax because it makes a discrete per-position scale and offset decision visible to downstream layers and to SpanPredictor, and it caps the per-position gradient and memory cost. To prevent a few motifs from monopolising the routing, we add a floor based load balancing term that penalises any candidate whose Top K utilisation falls below a target floor: a stop gradient sparse trigger pairs with a dense logarithmic gradient on the full softmax mass, so dead motifs continue to receive learning signal. This differs from the dot-product auxiliary load-balancing loss of Switch Transformers (Fedus et al., 2022); our dense-routing controls instead use a quadratic floor penalty (Appendix A.4).

3.2 SPANPREDICTOR

For each scale beyond the base level, a lightweight head (a linear projection followed by layer normalisation) maps the decoder hidden state into the dictionary space, and logits are formed by tying the head to the same beyond-token embedding used by SpanReader. Tying parameters across the input and output ports forces "which motif to inject" and "which k-mer to predict" to share a single representation, extending standard input–output weight tying (Press & Wolf, 2017; Baevski & Auli, 2019) to a multi scale dictionary in which the selector axis is the scale rather than the frequency band.

The first chain step predicts the next k-mer from the decoder state $\mathbf{h}^{(0)}$ directly. For longer-range supervision, we use teacher forcing: the next $n_{\max}$-mer target chunk is embedded via the tied dictionary and concatenated with the decoder state. The MLP MTP block (marked "Concat" in Fig. 2) maps this pair to a second decoder state for far-horizon supervision:

$$\begin{aligned} \mathbf{c}^{(0)} &= \mathbf{E}_{n_{\max}}[\text{chunk_id}], \\ \mathbf{h}^{(1)} &= \text{MTP}(\mathbf{h}^{(0)},\, \mathbf{c}^{(0)}). \end{aligned} \tag{3}$$

We deliberately omit self-attention inside the MTP block to keep future tokens from leaking from $\mathbf{c}^{(0)}_{t'>t}$ into $\mathbf{h}^{(1)}_t$. The second chain step then predicts a k-mer that begins after the chunk window, so the supervised target is token disjoint from the chunk embedding it conditions on.

Equal weighted cross-entropy across chain step and scale combinations lets short scales dominate the gradient, because their targets carry less entropy. We instead form a per-scale normalised loss $\ell_n = -\log p_n / \log(5^n)$ at each chain step and combine it with a per-scale length bonus β_n through a softmax gate over scales:

$$\begin{aligned} g_n &= \text{sg}[\text{softmax}_n((-\ell_n + \beta_n)/\tau)], \\ \mathcal{L}_{k\text{mer}} &= \left\langle \textstyle\sum_n g_n\, \ell_n \right\rangle_{c,t}. \end{aligned} \tag{4}$$

where $\text{sg}[\cdot]$ denotes a stop-gradient and $\langle\cdot\rangle_{c,t}$ averages over chain steps and over positions for which at least one scale has a valid target. The gate is detached on the forward path so it acts as a routing weight on the loss, not as another shortcut to be optimised; the bonus biases supervision toward longer motifs while keeping shorter scales responsive to CpG and codon level signals. The argmax of the gate is a useful analysis variable but is not part of the loss.

4 EXPERIMENTS

4.1 SETUP

We pretrain Beyond Tokens on an approximately 50 B token DNA corpus of human and nonhuman primate genomes. The human side draws one haplotype from each of 15 HPRC individuals (Liao et al., 2023) stratified across the five 1000 Genomes superpopulations (The 1000 Genomes Project Consortium, 2015) to balance ancestry coverage; the nonhuman side spans 9 primate species, with gene annotations harmonised across TOGA, NCBI EGAP/RefSeq, and UCSC xenoRefGene (Kirilenko et al., 2023; O'Leary et al., 2016; Nassar et al., 2023). Each genome is split into nonoverlapping 8,192 bp chunks, and we keep only chunks within 5 kb of an annotated gene to concentrate sampling on functionally informative regions; the human and primate halves are mixed at a 1:1 ratio at training time. We use only the forward strand and rely on the RC-aware decomposition of SpanReader for reverse complement coverage in place of data augmentation. The backbone is a 1.2 B sparse MoE Transformer. Full corpus construction, tokenizer conventions, and training hyperparameters are in Appendix A.

4.2 DOWNSTREAM TASKS

We evaluate pretrained representations on genomic tasks spanning gene structure, regulatory elements, variant effect, epigenetics, and species identity. The backbone is frozen, while a supervised probe is trained on labeled data for each task. We use the downstream probing protocol of Xu et al.

Table 1: Supervised frozen-representation probing (AUC). Comparison scores are from Table 3 of Xu et al. (2026). Best per row in **bold**; second best in *italics*.

Task	Beyond Tokens	Gengram	Genos	Evo2	Generator	NT-v3
Total params	1.2 B	10 B	10 B	40 B	3 B	650 M
Pretraining tokens	50 B	200 B	2.2 T	9.3 T	386 B	11 T
coding vs. intergenomic	0.9804	0.9733	**0.9914**	*0.9886*	0.9855	0.9830
splice sites	**0.9232**	0.9009	0.7990	*0.9138*	0.8071	0.8644
human genomic element	**0.9195**	0.8982	0.8890	0.9103	0.9173	*0.9189*
multi-species exon	0.9737	**0.9832**	*0.9755*	0.9332	0.9455	0.9615
promoter	**0.9342**	0.9074	*0.9249*	0.9227	0.9195	0.9220
enhancer	**0.7565**	0.7401	*0.7532*	0.7527	0.7390	0.7415
human enhancers (Cohn)	**0.8630**	0.8358	*0.8552*	0.7756	0.8181	0.8184
open chromatin	**0.7980**	*0.7714*	0.7623	0.7635	0.7270	0.7426
causal eQTL	*0.6928*	0.6518	0.6773	**0.7054**	0.6920	0.6870
CPC	0.9366	0.9219	**0.9522**	0.9401	0.9315	*0.9468*
H3	0.9371	*0.9434*	0.9400	0.9311	0.9163	**0.9505**
H3K36me3	0.8121	0.8039	0.7658	**0.8823**	0.8247	*0.8535*
species	0.9210	*0.9611*	**0.9739**	0.8867	0.8424	0.8652
Mean	**0.8806**	0.8686	0.8661	*0.8697*	0.8512	0.8658

(2026). The promoter, enhancer, and causal-eQTL datasets originate from the Genomics Long-Range Benchmark (LRB) (InstaDeepAI, 2024). Twelve tasks use a two-layer MLP and one uses gradient-boosted trees. Binary tasks are scored by ROC-AUC, the multi class species task by macro one-vs-rest AUC, and the reported task mean is the arithmetic mean across all task scores. Table 1 compares the full SpanReader + SpanPredictor model with Gengram (Xu et al., 2026), Genos (Lin et al., 2025), Evo2 (Brixi et al., 2026), Generator (Wu et al., 2025), and NT-v3 (Boshar et al., 2025). Method-level attribution is isolated by the matched-budget controls and ablations in Table 2.

Beyond Tokens achieves the highest task mean of 0.8806 in Table 1, with a 1.2 B model trained on approximately 50 B tokens. It exceeds the reported means of the 10 B Gengram and Genos models and the 40 B Evo2 model. It leads on six of the thirteen tasks, covering splice sites, human genomic elements, promoter and enhancer tasks, and open chromatin. Other models lead on coding-vs-intergenomic classification, multi-species exon detection, causal-eQTL prediction, coding potential, H3, H3K36me3, and species classification. The advantage is therefore in the task mean and a subset of tasks. These published-model comparisons provide benchmark context; direct evidence for the architectural changes comes from the matched-budget controls in Table 2. Figure 3 shows the corresponding per-task Full-versus-Baseline profile.

4.3 Ablation

Table 2 reports the within-budget Full-vs-Baseline progression, leave-one-out ablations, and sensitivity analyses. All configurations use the same 1.2 B backbone, approximately 50 B-token corpus, training budget, and downstream evaluation pipeline. The forward-addition rows add SpanReader and SpanPredictor to Baseline, whereas the remaining rows remove or alter one named component of Full.

Under the matched budget, SpanReader raises mean AUC from 0.8642 to 0.8752 (1.10 percentage points), while SpanPredictor reaches 0.8702 (0.60 points). Full reaches 0.8806 (1.64 points), close to the sum of the isolated gains.

Untying the dictionary lowers mean AUC to 0.8746, a 0.60-point reduction that directly supports parameter sharing in this setting. Removing the length-biased gate, RC-aware decomposition, Floor-LB, Chain MTP, or the length bonus lowers the mean by 0.86, 0.64, 0.41, 0.33, and 0.25 points, respectively. These fixed-seed comparisons are observed estimates rather than seed-averaged rankings (Appendix D).

Table 2: Matched-budget controls and ablations. Changes in task mean are in percentage points, relative to Baseline for forward addition and Full otherwise.

Configuration	Task mean	Change
Forward addition		
Baseline	0.8642	–
+ SpanReader only	0.8752	+1.10
+ SpanPredictor only	0.8702	+0.60
Full: SpanReader + SpanPredictor	**0.8806**	**+1.64**
Core component leave-one-out		
- disable RC-aware decomposition	0.8742	−0.64
- disable Chain MTP	0.8773	−0.33
- disable the length biased gate	0.8720	−0.86
- disable Floor-LB	0.8765	−0.41
- disable the length bonus β_n	0.8781	−0.25
- untie the shared dictionary	0.8746	−0.60
Hyperparameter / training technique sensitivity		
- enforce hard equivariance	0.8733	−0.73
- Floor-LB log-hybrid → quadratic	0.8794	−0.12
- disable multi scale ctx conv	0.8751	−0.55
- Top-K=4 → K=1	0.8654	−1.52
- Top-K=4 → K=8	0.8796	−0.10
- Top-K=4 → dense softmax	0.8766	−0.40

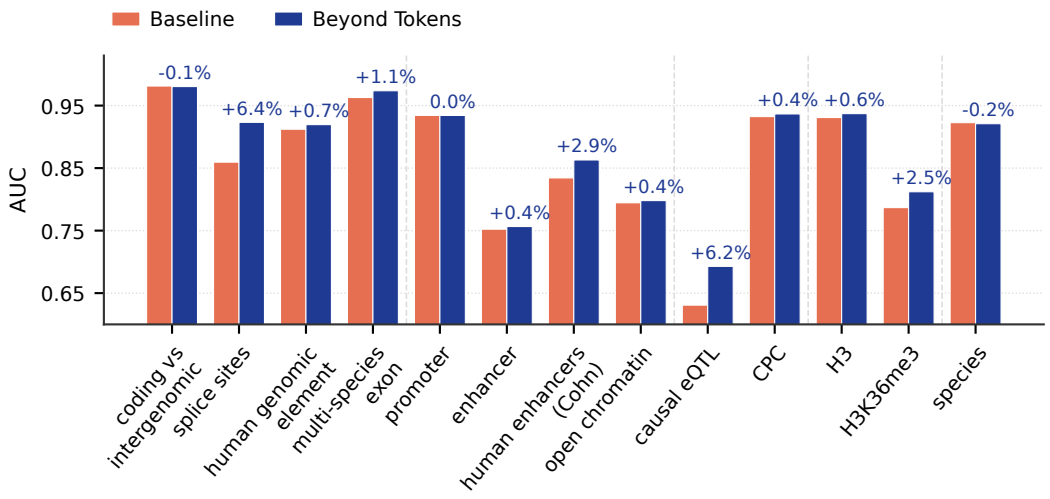


Figure 3: Per-task AUC for Beyond Tokens and the matched-budget Baseline. The Full-minus-Baseline gap is printed above each Beyond Tokens bar.

The sparsity sweep favours K=4: mean AUC falls to 0.8654 with K=1, 0.8796 with K=8, and 0.8766 with dense softmax. K=8 doubles the selected candidate count without improving the score; this comparison does not establish end-to-end training cost.

4.4 Interpretability

Figure 4 projects motifs from span lengths $n = 2, \ldots, 6$ into a shared UMAP coloured by motif category. The displayed neighbourhoods show category-associated structure that is not reducible to a single visible GC gradient. Figure 5 plots the asymmetric magnitude $\|\mathbf{m}_{fw} - \mathbf{m}_{rc}\|_2$ against a hexamer palindrome score, with a negative Spearman correlation in the inset; near-palindromic

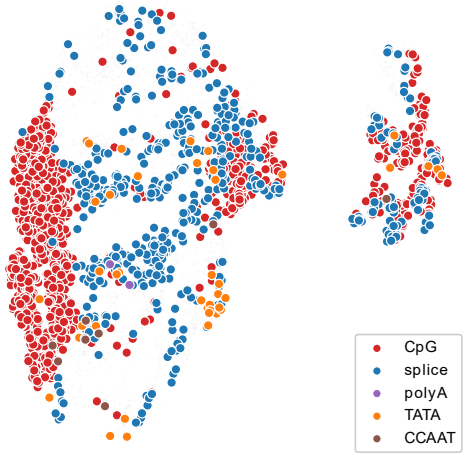


Figure 4: Dictionary UMAP across span lengths $n = 2 \ldots 6$, coloured by motif category.

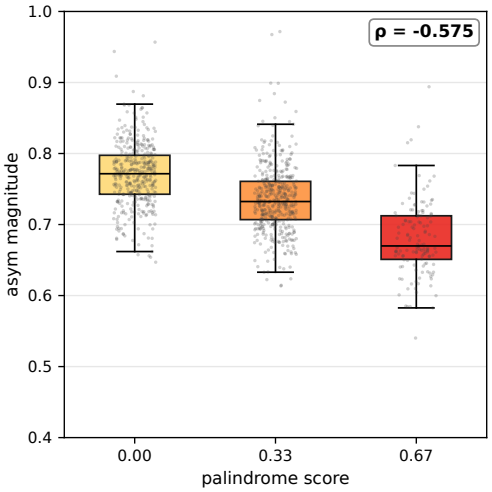


Figure 5: Asymmetric component magnitude versus palindrome score for hexamers; the inset reports Spearman's ρ.

hexamers receive small asymmetric capacity and strand-specific hexamers receive large capacity, so the model spends its asymmetric budget where RC symmetry is genuinely broken.

Figure 6 illustrates the display of span weights along an annotated 701 bp *ACTB* segment. Its enhanced peaks and randomised surrogate are illustrative and provide no evidence of motif enrichment or an untying effect. The supplementary package provides the numeric export and an unenhanced rendering; Appendix B describes the folding procedure.

Figure 7 compares n=6 projections under base concatenation and the learned dictionary. GC content dominates (a) but less visibly (c); named motif categories have higher two-dimensional silhouette in (d) than (b). These qualitative views do not independently validate high-dimensional clustering or explain the untying ablation.

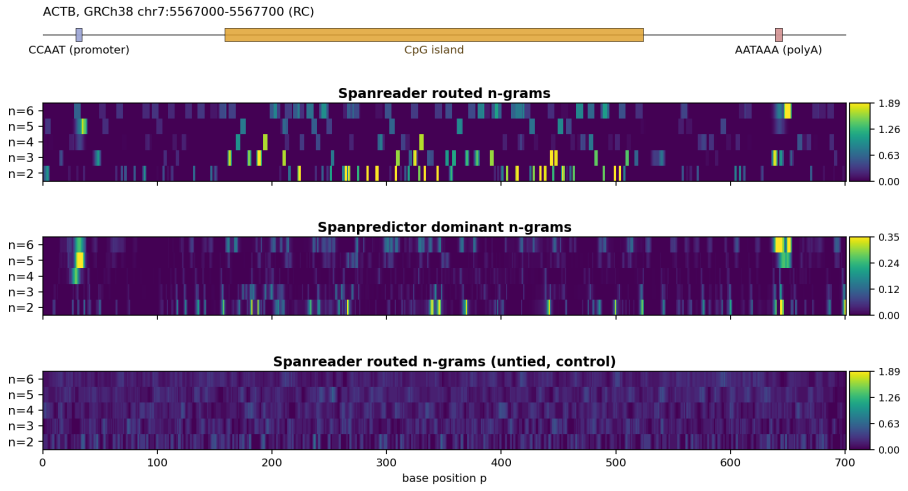


Figure 6: Illustrative rendering of an annotated *ACTB* segment. Regional amplification, synthetic peaks, and smoothing alter the exported weights. Colour-bar ticks show pre-enhancement ranges and cannot quantify the transformed values. The bottom panel, labelled "untied, control", is a randomised surrogate, not output from an untied model. An unenhanced view of the export is provided in the supplementary package.

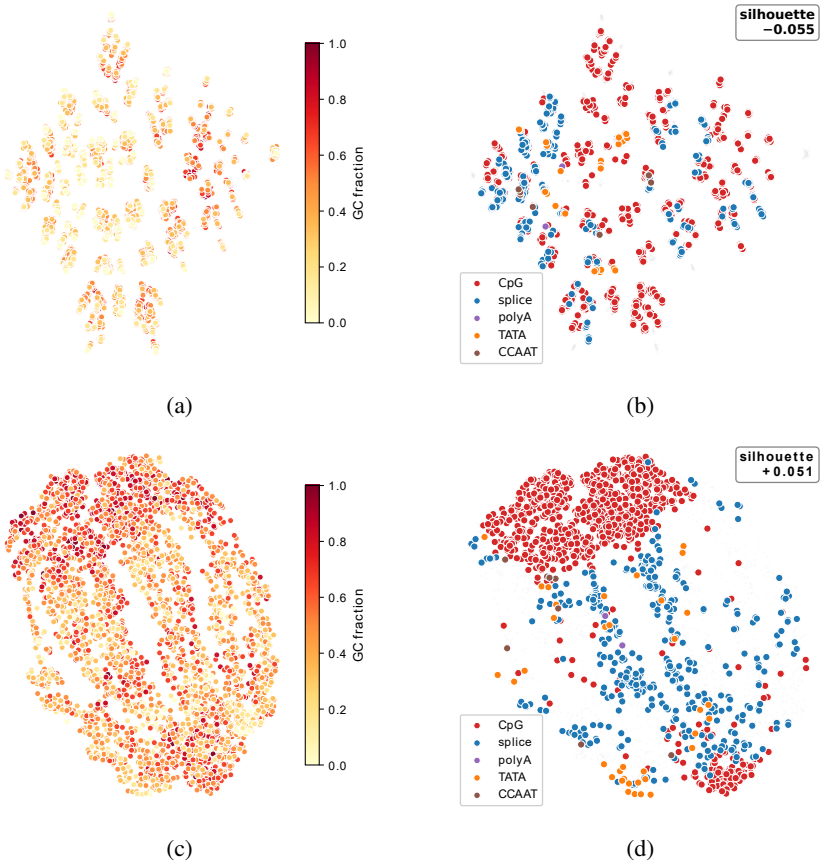


Figure 7: Hexamer projections under single-base concatenation (top) and the learned dictionary (bottom). Left panels are coloured by GC content; right panels by motif category, with the silhouette of named classes shown as inset.

5 Discussion and Limitations

Evaluation covers one 1.2 B sparse MoE backbone, approximately 50 B training tokens from human and nine nonhuman primate species, and one fixed-seed run per configuration. Broader corpora, additional backbones, and seed-averaged comparisons are needed to establish transfer. The enumerated dictionary relies on DNA's small alphabet; larger alphabets or longer spans would require factorised, hashed, or hierarchical dictionaries, which are not evaluated here.

The visualisations are descriptive, and frozen-representation benchmarks do not establish clinical validity. Appendix D.1 discusses the design trade-offs and responsible-use scope.

6 Conclusion

Beyond Tokens couples sparse context reading and atomic future-chunk prediction through shared multiscale dictionaries, preserving the base-level interface. In the evaluated genomic setting, it improves matched-budget mean AUC by 1.64 percentage points; a 0.60-point loss after untying supports dictionary sharing. The highest mean among the compared models provides complementary benchmark evidence. These results support the enumerated small-alphabet setting; transfer beyond it remains open. Future work includes length-adaptive decoding and causal motif interventions.

AI USE STATEMENT

Generative AI tools were used for language polishing. The authors take responsibility for the final content of the paper.

ETHICS STATEMENT

This study uses human and nonhuman primate genomic resources described in Appendix A.1; their use remains subject to the source terms and licences. The gene-enriched corpus and limited phylogenetic coverage constrain the scope of the empirical findings. Variant-effect evaluation is restricted to frozen-representation benchmarks and does not establish clinical validity. The model is not intended for diagnosis or treatment decisions. Further scope and responsible-use considerations appear in Appendix D.

REPRODUCIBILITY STATEMENT

Appendix A describes the pretraining corpus, tokenizer, backbone, forward pass, both modules, optimisation settings, and downstream probes. Table 2 specifies the matched-budget comparisons; Appendix C provides training diagnostics. The supplementary package includes the numeric export and plotting script for the unenhanced companion to Figure 6. Pretraining configurations are single fixed-seed runs, so the ablations are single-run estimates rather than seed-averaged rankings (Appendix D).

REFERENCES

Žiga Avsec, Vikram Agarwal, Daniel Visentin, et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18:1196–1203, 2021. doi: 10.1038/s41592-021-01252-x.

Žiga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle R. Taylor, Tom Ward, Clare Bycroft, Lauren Nicolaisen, Eirini Arvaniti, Joshua Pan, Raina Thomas, Vincent Dutordoir, Matteo Perino, Soham De, Alexander Karollus, Adam Gayoso, Toby Sargeant, Anne Mottram, Lai Hong Wong, Pavol Drotár, Adam Kosiorek, Andrew Senior, Richard Tanburn, Taylor Applebaum, Souradeep Basu, Demis Hassabis, and Pushmeet Kohli. Advancing regulatory variant effect prediction with AlphaGenome. *Nature*, 649:1206–1218, 2026. doi: 10.1038/s41586-025-10014-0.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1809.10853.

Gonzalo Benegas, Carlos Albors, Alan J. Aw, Chengzhong Ye, and Yun S. Song. A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, 43:1960–1965, 2025. doi: 10.1038/s41587-024-02511-w.

Sam Boshar, Benjamin Evans, Ziqi Tang, Armand Picard, Yanis Adel, Franziska K. Lorbeer, Chandana Rajesh, Tristan Karch, Shawn Sidbon, David Emms, Javier Mendoza-Revilla, Fatimah Al-Ani, Evan Seitz, Yair Schiff, Yohan Bornachot, Ariana Hernandez, Marie Lopez, Alexandre Laterre, Karim Beguir, Peter Koo, Volodymyr Kuleshov, Alex Stark, Bernardo P. de Almeida, and Thomas Pierrot. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. *bioRxiv*, 2025. doi: 10.64898/2025.12.22.695963. URL https://doi.org/10.64898/2025.12.22.695963.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652:1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Yilong Chen, Yanxi Xie, Zitian Gao, He Xin, Yihao Xiao, Jason Klein Liu, Haoming Luo, Yifan Luo, Zhengmao Ye, Tingwen Liu, Xin Zhao, Ran Tao, and Bryan Dai. Beyond n-gram: Data-aware X-GRAM extraction for efficient embedding parameter scaling. *arXiv preprint arXiv:2604.21724*, 2026. URL https://arxiv.org/abs/2604.21724.

Xin Cheng, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Zhewen Hao, Yukun Li, Han Zhang, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models. *arXiv preprint arXiv:2601.07372v1*, 2026. URL https://arxiv.org/abs/2601.07372v1.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Björn Deiseroth, Manuel Brack, Patrick Schramowski, Kristian Kersting, and Samuel Weinbach. T-FREE: Subword tokenizer-free generative LLMs via sparse representations for memory-efficient embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21829–21851. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1217. URL https://aclanthology.org/2024.emnlp-main.1217/.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://www.jmlr.org/papers/v23/21-0998.html. arXiv:2101.03961.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15706–15734. PMLR, 2024. URL https://proceedings.mlr.press/v235/gloeckle24a.html.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Zhou Xun. Over-tokenized transformer: Vocabulary is generally worth scaling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26261–26282. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25bb.html.

Human Pangenome Reference Consortium. HPRC release 2 assembly resource and data tables. Official assembly resource documentation, 2025. URL https://github.com/human-pangenomics/hprc_intermediate_assembly/blob/main/data_tables/README.md. Accessed 2026-09-26.

InstaDeepAI. Genomics Long-Range Benchmark. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/InstaDeepAI/genomics-long-range-benchmark.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V. Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. doi: 10.1093/bioinformatics/btab083.

Bogdan M. Kirilenko, Chetan Munegowda, Ekaterina Osipova, et al. Integrating gene annotation with orthology inference at scale. *Science*, 380(6643):eabn3107, 2023. doi: 10.1126/science.abn3107. URL https://doi.org/10.1126/science.abn3107.

Qiuyi Li, Zhihao Zhan, Shikun Feng, Yiheng Zhu, Yuan He, Wei Wu, Zhenghang Shi, Shengjie Wang, Zongyong Hu, Zhao Yang, Jiaoyang Li, Jian Tang, Haiguang Liu, and Tao Qin. GENERator-v2: Reconciling coarse tokenization with single-nucleotide resolution in genomic language modeling. *bioRxiv*, 2026. doi: 10.64898/2026.01.27.702015. URL https://doi.org/10.64898/2026.01.27.702015.

Siyuan Li, Zedong Wang, Zicheng Liu, Di Wu, Cheng Tan, Jiangbin Zheng, Yufei Huang, and Stan Z. Li. VQDNA: Unleashing the power of vector quantization for multi-species genomic sequence modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28717–28733. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bm.html.

Wen-Wei Liao, Mobin Asri, Jana Ebler, et al. A draft human pangenome reference. *Nature*, 617 (7960):312–324, 2023. doi: 10.1038/s41586-023-05896-x. URL https://doi.org/10.1038/s41586-023-05896-x.

Adi Lin et al. Genos: a human-centric genomic foundation model. *GigaScience*, 2025. doi: 10.1093/gigascience/giaf132.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R. Kelley. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *Nature Genetics*, 57:949–961, 2025. doi: 10.1038/s41588-024-02053-6.

Hong Liu, Jiaqi Zhang, Chao Wang, Xing Hu, Linkun Lyu, Jiaqi Sun, Xurui Yang, Bo Wang, Fengcun Li, Yulei Qian, Lingtong Si, Yerui Sun, Rumei Li, Peng Pei, Yuchen Xie, and Xunliang Cai. Scaling embeddings outperforms scaling experts in language models. *arXiv preprint arXiv:2601.21204*, 2026. URL https://arxiv.org/abs/2601.21204.

Mingqian Ma. Reverse-complement consistency for DNA language models. *arXiv preprint arXiv:2509.18529*, 2025. URL https://arxiv.org/abs/2509.18529.

Vincent Mallet and Jean-Philippe Vert. Reverse-complement equivariant networks for DNA sequences. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13511–13523. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/706608cfdbcc1886bb7eea5513f90133-Abstract.html.

Lei Mao, Yuanhe Tian, and Yan Song. DNAZEN: Enhanced gene sequence representations via mixed granularities of coding units. *arXiv preprint arXiv:2505.02206*, 2025. URL https://arxiv.org/abs/2505.02206.

Frederikke I. Marin, Felix Teufel, Marc Horlacher, Dennis Madsen, Dennis Pultz, Ole Winther, and Wouter Boomsma. BEND: Benchmarking DNA language models on biologically meaningful tasks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/429e7b31625a8b7839f9e4d6e2aa9bb9-Abstract-Conference.html. arXiv:2311.12570.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pp. 246–252. PMLR, 2005. URL https://proceedings.mlr.press/r5/morin05a.html.

Luis R. Nassar, Galt P. Barber, Anna Benet-Pagès, et al. The UCSC genome browser database: 2023 update. *Nucleic Acids Research*, 51(D1):D1188–D1195, 2023. doi: 10.1093/nar/gkac1072. URL https://doi.org/10.1093/nar/gkac1072.

Eric Nguyen, Michael Poli, Marjan Faizi, et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. In *NeurIPS*, 2023. arXiv:2306.15794.

Nuala A. O'Leary, Mathew W. Wright, J. Rodney Brister, et al. Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, 2016. doi: 10.1093/nar/gkv1189. URL https://doi.org/10.1093/nar/gkv1189.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163. Association for Computational Linguistics, 2017. URL https://aclanthology.org/E17-2025/.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. ProphetNet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2401–2410. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.217. URL https://aclanthology.org/2020.findings-emnlp.217/.

Lifeng Qiao, Peng Ye, Yuchen Ren, Weiqiang Bai, Chaoqi Liang, Xinzhu Ma, Nanqing Dong, and Wanli Ouyang. Model decides how to tokenize: Adaptive DNA sequence tokenization with MxDNA. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66080–66107, 2024. doi: 10.52202/079017-2112. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/79af547fa22cdcb0facd0b31dcd4bdb0-Abstract-Conference.html.

Melissa Sanabria, Jonas Hirsch, Pierre Joubert, and Anna R. Poetsch. DNA language model GROVER learns sequence context in the human genome. *Nature Machine Intelligence*, 6:911–923, 2024. doi: 10.1038/s42256-024-00872-0.

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range DNA sequence modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43632–43648. PMLR, 2024. URL https://proceedings.mlr.press/v235/schiff24a.html.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv*, 2017. doi: 10.1101/103663. URL https://doi.org/10.1101/103663.

Ziqi Tang, Nirali Somia, YiYang Yu, and Peter K. Koo. Evaluating the representational power of pre-trained DNA language models for regulatory genomics. *Genome Biology*, 26(1):203, 2025. doi: 10.1186/s13059-025-03674-8.

The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526:68–74, 2015. doi: 10.1038/nature15393. URL https://doi.org/10.1038/nature15393.

Wei Wu, Qiuyi Li, Yuanyuan Zhang, Zhihao Zhan, Ruipu Chen, Mingyang Li, Kun Fu, Junyan Qi, Yongzhou Bao, Chao Wang, Yiheng Zhu, Zhiyun Zhang, Jian Tang, Fuli Feng, Jieping Ye, Yuwen Liu, Hui Xiong, and Zheng Wang. GENERator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

Huinan Xu, Xuyang Feng, Junhong Chen, Junchen Liu, Kaiwen Deng, Kai Ding, Shengning Long, Jiaxue Shuai, Zhaorong Li, Shiping Liu, Guirong Xue, and Zhan Xiao. Beyond conditional computation: Retrieval-augmented genomic foundation models with gengram. *arXiv preprint arXiv:2601.22203*, 2026. URL https://arxiv.org/abs/2601.22203.

Xiaoxiao Zhou, Zihan Wang, Jingbo Shang, and Yang E. Li. DNAMotifTokenizer: Towards biologically informed tokenization of genomic sequences. *arXiv preprint arXiv:2512.17126*, 2025. URL https://arxiv.org/abs/2512.17126.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V. Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *ICLR*, 2024. arXiv:2306.15006.

A Reproducibility Details

A.1 Pretraining data

The pretraining corpus contains approximately 50 B tokens and has two components. **Human:** from Human Pangenome Reference Consortium (HPRC) release 2 (Liao et al., 2023; Human Pangenome Reference Consortium, 2025), we sample one haplotype from each of 15 individuals stratified across the five 1000 Genomes superpopulations (AFR, EAS, AMR, SAS, and EUR) (The 1000 Genomes Project Consortium, 2015). Gene annotations are filtered with CAT v1.3 GFF3 files. **Nonhuman primates:** we use 9 species, with annotations harmonised across TOGA (Kirilenko et al., 2023), NCBI EGAP/RefSeq (O'Leary et al., 2016), and UCSC xenoRefGene (Nassar et al., 2023).

Each genome is divided into nonoverlapping 8,192 bp chunks. We retain chunks whose nearest annotated gene lies within 5,120 bp (approximately 5 kb) and mix the human and nonhuman components at an approximately 1:1 ratio. This sampling concentrates the corpus around gene-associated regions; the main paper therefore separates cross-model comparisons from matched-corpus ablations. Pretraining uses the forward strand only. Reverse-complement structure is supplied by the RC-aware decomposition in SpanReader rather than by data augmentation.

The tokenizer follows the Hugging Face single-base convention used by the backbone: A, C, G, T, N, and EOD have token IDs $8, 5, 6, 7, 9$, and 12, respectively. Tokens outside A/C/G/T/N are masked during motif construction.

A.2 Backbone architecture

The model uses the 1.2 B-scale sparse-MoE Transformer backbone described in the main paper: 12 layers, hidden size $H{=}1024$, 16 attention heads with 8 GQA query groups, and 8 routed experts under Top-2 token-choice routing. Each expert has a 4096-wide SwiGLU feed-forward block. The model uses RMSNorm, RoPE with base 10^4, bf16 computation, and a maximum sequence length of 8192. Attention uses a local window $(511, 0)$ in the configured sliding-window schedule, with selected layers retaining global attention. MoE routing uses an auxiliary coefficient of 10^{-3}, z-loss coefficient of 10^{-3}, fp32 router computation, token all-to-all dispatch, and grouped GEMM.

A.3 Forward pass

Table 3 summarises one training forward pass. The base-level language-model path is unchanged; Beyond Tokens contributes a residual input injection and an auxiliary chunk-level loss.

Table 3: Beyond Tokens forward pass for one training batch.

Input: token IDs $\mathbf{x} \in \mathcal{V}^S$ and shared dictionary $\{\mathbf{E}_n\}_{n=1}^{6}$. **Output:** hidden states $\mathbf{h}$ and loss $\mathcal{L}$.
Input port: SpanReader
1 Compute the base embedding $\mathbf{e} = \mathbf{E}_w[\mathbf{x}]$ and the normalised routing query $\widetilde{\mathbf{Q}}$ from causal multiscale convolutions of $\mathbf{e}$.
2 Enumerate the valid scale–offset candidates and construct their RC-aware keys and values from the shared dictionary.
3 Apply sparse Top-K routing to obtain $\mathbf{u}$; run the backbone on $\mathbf{e}' = \mathbf{e} + \mathbf{u}$ to obtain $\mathbf{h}^{(0)}$.
Output port: SpanPredictor
4 Form the teacher-forced chunk embedding $\mathbf{c}^{(0)} = \mathbf{E}_{n_{\max}}[\text{chunk_id}]$ and compute $\mathbf{h}^{(1)} = \mathrm{MTP}(\mathbf{h}^{(0)}, \mathbf{c}^{(0)})$.
5 For $c \in \{0, 1\}$ and $n \in \{2, \ldots, 6\}$, compute $\mathrm{logits}_n^{(c)} = \mathrm{Head}_n(\mathbf{h}^{(c)})\mathbf{E}_n^\top$ and aggregate the normalised chunk losses with the length-biased gate.
6 Return $\mathcal{L} = \mathcal{L}_{\mathrm{LM}} + w_{\mathrm{aux}}\mathcal{L}_{k\mathrm{mer}} + w_{\mathrm{lb}}\mathcal{L}_{\text{floor-LB}}$.

A.4 SpanReader implementation

Candidate construction and query path. SpanReader is applied immediately before the first Transformer layer and lifts the base embedding through the direct residual $\mathbf{e}' = \mathbf{e} + \mathbf{u}$. Within each W=21 bp window, we enumerate J=111 candidates (n, d) with $n \in \{1, \dots, 6\}$ and $d \in \{0, \dots, W - n\}$. All candidates are causal with respect to position t. For $n \geq 2$, the candidate is base-5 encoded, looked up in $\mathbf{E}_n$, and projected by $\mathbf{P}_n$. The query is formed by concatenating three depthwise causal 1D convolutions with kernel widths $\{21, 128, 512\}$, applying $\mathbf{W}_{\text{ctx}}$ and $\mathbf{W}_q$, and L_2 normalising along the hidden dimension. The key side is not normalised, so motif magnitude remains available to the router.

RC-aware keys. For each $n \geq 2$ candidate, we retrieve the forward and reverse-complement embeddings and compute

$$\begin{aligned} \text{sym} &= \tfrac{1}{2}(\mathbf{m}_{fw} + \mathbf{m}_{rc}), \\ \text{asym} &= \|\mathbf{m}_{fw} - \mathbf{m}_{rc}\|_2 \, \mathbf{D}_n[\text{fw_id}]. \end{aligned} \tag{5}$$

The learnable direction table $\mathbf{D}_n$ is initialised with standard deviation $0.02/\sqrt{H}$. We do not enforce $\mathbf{D}_n[\text{rc}] = -\mathbf{D}_n[\text{fw}]$, yielding the partial rather than strict RC equivariance evaluated in the main-paper ablation. Length and offset biases are applied as scalar score contributions, avoiding repeated materialisation of a $J \times H$ bias tensor at each position.

Sparse routing and floor load balancing. The routing temperature is parameterised as $\theta_{\text{scale}} = \text{softplus}(\theta') + \epsilon$ and initialised near 20, ensuring a positive scale throughout training. Invalid candidates are assigned a large negative mask before Top-K selection. With K=4, the selected scores are softmax-normalised and mixed through the value projection; positions with no valid candidate are explicitly zeroed.

The default floor-based load-balancing term is

$$\mathcal{L}_{\text{floor-LB}} = -\sum_{j=1}^{J} \text{ReLU}\big(\rho/J - f^{\text{sg}}_{\text{route},j}\big) \log(f_{\text{full},j} + \varepsilon)\,, \tag{6}$$

where the stop-gradient sparse frequency triggers the penalty and the full-softmax frequency supplies the gradient. Thus, candidates not selected by Top-K still receive signal. The dense-softmax and scale-only controls use the corresponding quadratic floor penalty because sparse and full frequencies coincide in those settings. The floor ratio is ρ=0.5.

Compute path. Motif materialisation uses a fused lookup-and-bias path over the 21 bp window. The value projection is applied once to the stacked multiscale tensor before gathering the Top-K indices, rather than as separate small matrix multiplications for every span length.

A.5 SpanPredictor implementation

Tied chunk heads. For each $n \in \{2, 3, 4, 5, 6\}$, the chunk head is $\text{LayerNorm}(\text{Linear}_{H \to H}(\mathbf{h}))$. Its logits are obtained through multiplication by $\mathbf{E}_n^\top$, tying the output classifier to the dictionary read by SpanReader.

Chain MTP. Chain step c=0 uses the decoder state $\mathbf{h}^{(0)}$ directly. Step c=1 uses $\mathbf{h}^{(1)} = \text{MTP}(\mathbf{h}^{(0)}, \mathbf{c}^{(0)})$, where $\mathbf{c}^{(0)}_{b,t}$ is the per-position teacher-forced embedding of the next $n_{\max}$=6 bases. The MTP block first forms

$$\mathbf{z} = \mathbf{W}_f[\text{LN}_h(\mathbf{h}^{(0)}) \| \text{LN}_c(\mathbf{c}^{(0)})] \tag{7}$$

and then applies the residual MLP

$$\begin{aligned} \text{MTP}(\mathbf{z}) &= \mathbf{z} + \mathbf{W}_2 \text{GELU}(\mathbf{W}_1 \text{LN}(\mathbf{z})), \\ \mathbf{W}_1 &\in \mathbb{R}^{4H \times H}, \qquad \mathbf{W}_2 \in \mathbb{R}^{H \times 4H}. \end{aligned} \tag{8}$$

The block contains no self-attention. The step-1 target begins after the conditioning chunk, at $\mathbf{x}[t + n_{\max} + 1 : t + n_{\max} + n + 1]$, and is therefore token-disjoint from the window used to form $\mathbf{c}^{(0)}$. Validity masks are accumulated across chain steps to keep every target in bounds.

Length-biased confidence gate. At each chain step, the per-scale loss is normalised as $\ell_n = -\log p_n / \log(5^n)$, combined with bonus β_n, and passed through a stop-gradient softmax with temperature τ. This gate weights the loss but is not itself optimised as a shortcut. Its argmax is recorded only as a dominant-scale analysis variable.

A.6 Optimisation and training setup

Training uses sequence length 8192, global batch size 1152, microbatch size 6, and gradient accumulation 6. The schedule contains $5{,}762{,}773$ training samples, corresponding to approximately $5{,}002$ optimisation steps and 47.21 B consumed tokens; the main paper reports this budget as approximately 50 B tokens. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, gradient clipping at 1.0, and parameter initialisation standard deviation 0.01. The learning rate follows cosine decay from 10^{-4} to 10^{-5}, with a 5% warmup and decay over 95% of training samples.

The main model and every pretraining ablation are single runs under a fixed seed; the main-model seed is 42, and no seed search is performed. Training runs on 4 nodes with 8 NVIDIA A800-80GB GPUs per node using Megatron-LM/PyTorch, with TP$=1$, PP$=1$, and EP$=1$. Computation is bf16; attention softmax and gradient all-reduce use fp32, and bf16 reduced-precision matrix multiplication is disabled. Megatron's auxiliary-loss autoscaler attaches $w_{\text{aux}}\mathcal{L}_{k\text{mer}} + w_{\text{lb}}\mathcal{L}_{\text{floor-LB}}$ to the backward graph without changing forward activations.

Table 4: Default hyperparameters introduced by Beyond Tokens. They are shared across the main model and matched-budget controls unless a named sensitivity row in the main paper changes the corresponding setting.

Hyperparameter	Value	Role
w_{aux}	0.5	weight of $\mathcal{L}_{k\text{mer}}$
w_{lb}	0.01	weight of $\mathcal{L}_{\text{floor-LB}}$
ρ	0.5	per-candidate activation floor ratio
τ	0.5	temperature of the length-biased gate
$(\beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$	$(0.3, 0.3, 0.4, 0.4, 0.6)$	per-scale length bonus
K	4	candidates retained per position
D	1	MTP depth; two supervised chain steps
θ_{scale} initial value	approximately 20	routing temperature
W / J	21 bp / 111	routing window / scale–offset candidates

A.7 Downstream evaluation protocol

The backbone is frozen, and the decoder hidden states after the final layer normalisation are mean-pooled over valid sequence positions; EOD and padding positions are excluded. This pooling rule was selected once in a four-task pilot comparing final-layer mean pooling, multilayer fusion, and last-token pooling, and was then held fixed for all Beyond Tokens variants and matched-budget ablations. Twelve binary tasks use scikit-learn's `MLPClassifier` with hidden sizes $(256, 128)$, early-stopping patience 10, maximum 500 iterations, and probe seed 0. The species task uses gradient-boosted trees with 500 estimators, maximum depth 8, learning rate 0.1, and seed 0. Binary tasks use ROC-AUC; species classification uses macro one-vs-rest AUC. The task mean is the unweighted arithmetic mean of all 13 scores.

All Beyond Tokens variants and matched-budget ablations use this same local feature-extraction and probing pipeline.

A.8 Module ownership

Table 6 makes the two-port ownership explicit. The shared dictionaries participate in both ports: SpanReader reads them as candidate memory, whereas SpanPredictor uses them as tied classifier bases. RC-aware keys, routing queries, values, and multiscale context belong to SpanReader; chunk heads, Chain MTP, and the length-biased gate belong to SpanPredictor. We do not infer separate parameter totals from rounded model-size reporting in the main paper.

Table 5: The 13 downstream tasks used in the main paper. Cross-model values are reported from Xu et al. (2026); component attribution uses only matched-budget ablations. The promoter, enhancer, and causal-eQTL tasks use LRB datasets (InstaDeepAI, 2024).

Group	Task in main paper	Internal task identifier	Probe	Metric
Gene structure	coding vs. intergenomic	`demo_coding_vs_intergenomic`	MLP	ROC-AUC
	splice sites	`splice_sites_all`	MLP	ROC-AUC
	human genomic element	`human_genomic_element`	MLP	ROC-AUC
	multi-species exon	`multi_species_exon`	MLP	ROC-AUC
Regulation	promoter	`regulatory_promoter_8k`	MLP	ROC-AUC
	enhancer	`regulatory_enhancer_8k`	MLP	ROC-AUC
	human enhancers (Cohn)	`human_enhancers_cohn`	MLP	ROC-AUC
	open chromatin	`human_ocr_ensembl`	MLP	ROC-AUC
Variant/ coding	causal eQTL	`variant_effect_eqtl_8k`	MLP	ROC-AUC
	CPC	`CPC_8192`	MLP	ROC-AUC
Epigenetics	H3	`H3`	MLP	ROC-AUC
	H3K36me3	`H3K36me3`	MLP	ROC-AUC
Evolution	species	`primate_species_8192`	boosted trees	macro OvR AUC

Table 6: Ownership of the modules in the complete two-port design.

Component	SpanReader	SpanPredictor
Shared dictionaries $\{\mathbf{E}_n\}$	read	tied logits
Scale projections $\{\mathbf{P}_n\}$	candidate keys	—
RC direction tables $\{\mathbf{D}_n\}$	RC-aware keys	—
Context convolutions	routing query	—
Query/value projections and biases	routing/residual	—
Chunk heads	—	chunk logits
Chain-MTP block	—	horizon 2
Length-biased gate	—	loss weighting

B Additional Routing Analysis

The unenhanced companion to Figure 6, supplied in the supplementary package, uses the available 701 bp export and excludes padded queries. Span weights are divided by n and accumulated over covered bases; horizon 2 is offset by six bases. Gate panels share a linear colour scale; routing uses its own. Unlike the illustrative figure, this companion applies no smoothing, regional amplification, synthetic peaks, or randomised control. No actual untied-model export is available.

C Training Diagnostics

Figure 8 shows the full training trajectory of the checkpoint used in the main paper. It monitors whether all chunk scales remain active, whether the second MTP horizon learns at comparable difficulty to the first, and whether optimisation remains stable.

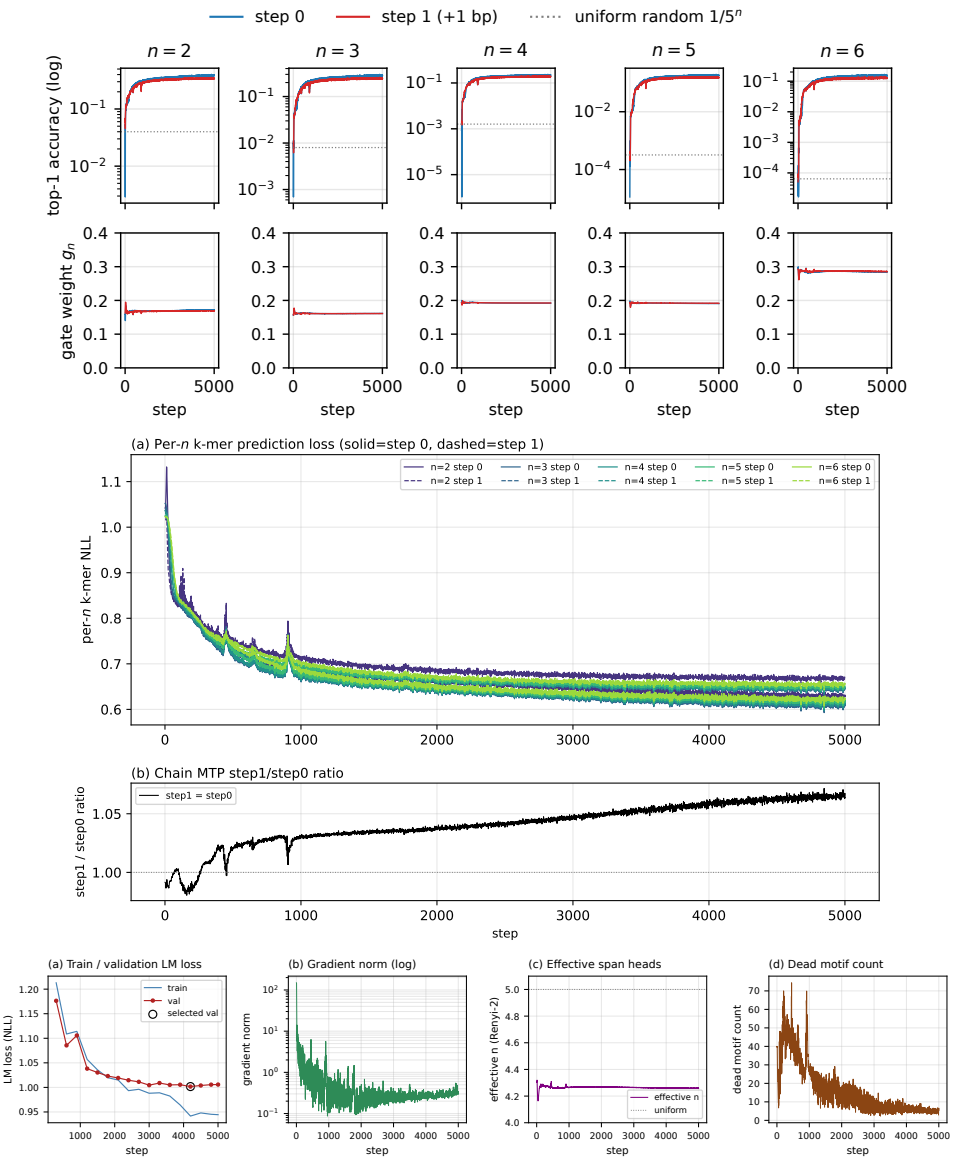


Figure 8: Training diagnostics. Top: per-scale chunk top-1 accuracy and length-biased gate weight g_n for both chain steps; dotted grey marks the uniform random baseline $1/5^n$. Middle: per-scale chunk NLL and the aggregate step-1/step-0 NLL ratio. Bottom: language-model loss, gradient norm, effective number of active scale heads $1/\sum_n g_n^2$, and candidate count below the floor-utilisation threshold. The plotted trajectories indicate that all scales remain active, both chain horizons improve, and optimisation remains stable; the circled validation point is the checkpoint used for downstream probing.

D SCOPE, DATA, AND RESPONSIBLE USE

The empirical scope is single-base autoregressive pretraining of one 1.2 B-scale sparse-MoE model on a gene-enriched human/primate corpus. The matched controls isolate architectural changes under that corpus and training budget; they do not establish that the same effect size holds under uniformly sampled whole genomes, broader phylogenetic coverage, longer-context backbones, or other alphabet sizes. Each pretraining configuration uses one prespecified fixed-seed run; the matched ablations are therefore interpreted as single-run estimates rather than seed-averaged rankings.

The variant-effect task is a benchmark of frozen representations, not evidence for clinical use. The model is not intended for diagnosis or treatment decisions. Data are obtained from the cited HPRC, 1000 Genomes, TOGA, NCBI/RefSeq, and UCSC resources and remain subject to their source terms and licences. The release accompanying an accepted paper will document preprocessing and model checkpoints needed to reproduce the reported pipeline.

D.1 EXTENDED DISCUSSION AND DESIGN TRADE-OFFS

Dual role of the tied dictionary. The same $\{\mathbf{E}_n\}_{n=1}^{6}$ table serves as both an input side motif dictionary read by sparse routing in SpanReader and an output side classifier predicted by chunk level heads in SpanPredictor: SpanReader optimises which motif is selected, SpanPredictor optimises which k-mer the hidden state can decode, and the two gradient streams meet on the dictionary. Untying it causes a drop in Table 2, consistent with read–write coupling.

Scope and design trade-offs. The study is scoped to single-base autoregressive pretraining of a 1.2 B sparse MoE backbone, under a fixed compute budget in Table 2. Thus, computational constraints limited the controlled pretraining runs to approximately 50 B tokens from human and nine nonhuman primate species; evaluation with larger and more phylogenetically diverse corpora remains future work. On RC equivariance, sym and the asymmetric magnitude are invariant while $\mathbf{D}_n[\mathrm{fw_id}]$ retains strand-specific freedom; our strict-equivariance ablation performs slightly worse than Full (Table 2). The hexamer chunk space already approaches full enumeration; longer spans could use hashed embeddings or a hierarchical softmax (Morin & Bengio, 2005). The current code path is single-stage; multi-stage pipeline parallelism would need a cross-stage all-reduce of the tied dictionary.

Implications for genomic foundation models. Fixed k-mer and BPE tokenizers define input units before pretraining (Ji et al., 2021; Zhou et al., 2024). Beyond Tokens preserves base alignment while selecting multiscale spans per position and supervising atomic chunk targets. Sharing input and output dictionaries, standard in NLP (Press & Wolf, 2017) but rarely seen for DNA, is supported by the untying ablation. The exact parameterisation studied here depends on DNA's small alphabet: its $O(|\Sigma|^n)$ dictionary remains enumerable through $n{=}6$. Larger alphabets would require factorised, hashed, or hierarchical dictionaries, which we do not evaluate. We thus claim generality for the read–predict tying principle, not for the present enumerated implementation.

Broader impact, future work. We study pretraining inductive biases. Under the same experimental setup, the Full-versus-Baseline ablation shows gains in frozen-representation probing on several noncoding regulatory, splicing, and variant-effect tasks; the risk is downstream use without sufficient validation. We confine the variant-effect analysis to benchmark probing and do not advocate clinical deployment; the pretraining data sources are listed in Appendix A.1. Follow-up directions: lifting the per position dominant scale from a training time analysis quantity to an inference time length adaptive decoding protocol, and porting the dual port design to backbones such as Caduceus and Evo 2.