GeneZip: Region-Aware Compression for Long-Context DNA Modeling

Jianan Zhao[1,2,*], **Xixian Liu**[1,2,*], **Zhihao Zhan**[1,2], **Xinyu Yuan**[1,2], **Hongyu Guo**[3,4], **Jian Tang**[1,2,5]
[1]Mila - Québec AI Institute [2]Université de Montréal
[3]National Research Council of Canada [4]University of Ottawa [5]HEC Montréal

Abstract

Long-context DNA models are limited by token-mixing cost and by how compression allocates representational budget across the genome. Existing approaches operate close to base-pair resolution, apply fixed downsampling, or learn content-dependent chunks without an explicit genomic budget, making long-context pretraining expensive and difficult to control. We introduce GeneZip, a region-aware DNA compression framework that combines H-Net-style dynamic routing with a Region-Aware Ratio (RAR) objective and bounded routing. GeneZip uses static gene-structure annotations during compression training to specify region-wise base-pairs-per-token (BPT) targets; at inference time, it compresses raw unseen DNA without annotations. GeneZip provides three main benefits. First, it is effective: GeneZip variants achieve the best validation PPL among encoder-based compressors, with GeneZip-70M already operating at 137.6 BPT, and across four reproducible DNALongBench tasks—contact map prediction, eQTL prediction, enhancer–target gene prediction, and transcription-initiation signal prediction—GeneZip obtains the best average rank among compared sequence models. Second, it is redundancy-aware: a post-hoc RepeatMasker/TRF analysis shows that, without repeat supervision, GeneZip assigns higher local BPT to TE-derived interspersed repeats and tandem repeats, two major classes of repetitive DNA sequence redundancy. Third, it is efficient: by reducing the effective token-mixing length, GeneZip enables longer-context and larger-capacity pretraining, including 128K-context and 636M-parameter variants on a single A100 80GB GPU, and fine-tunes the eQTL task 50.4× faster than JanusDNA (50 vs. 2520 minutes). These results establish GeneZip as an effective, redundancy-aware, and efficient compression interface for long-context DNA modeling.

1 Introduction

Long-context DNA models are increasingly used for sequence-to-function prediction, variant effect scoring, and 3D genome modeling, yet tasks requiring hundreds of kilobases to megabase-scale context remain bottlenecked by token mixing over very long sequences [1–3]. Current approaches reduce model cost with efficient long-sequence backbones or smaller capacity [4–6], or shorten the sequence through uniform U-Net-style downsampling [1, 2]. These strategies make long inputs feasible while allocating resolution according to architecture instead of genome structure. DNA violates this uniformity assumption: promoters, exons, UTRs, introns, and intergenic regions have different information densities; mammalian genomes contain abundant repetitive sequence redundancy, especially TE-derived interspersed repeats and tandem repeats; and the compartments that matter most differ across expression, enhancer–gene, and chromatin-structure tasks [7–9].

[*]Equal contribution. Correspondence to Jian Tang. Project page: `https://github.com/AndyJZhao/GeneZip`.

Preprint.

Motivated by this imbalance, we propose **GeneZip**, a region-adaptive compression module that learns non-uniform token allocation over ultra-long DNA. GeneZip provides a compression interface for long-context DNA modeling: it builds on H-Net-style dynamic chunking and routing [10], while adding biological control over where the token budget is spent. Specifically, its Region-Aware Ratio (RAR) objective trains the router toward region-wise base-pairs-per-token (BPT) targets derived from static gene-structure annotations, and bounded routing enforces a global token budget for stable long-context training. The region annotations are used only during compression training, so GeneZip compresses unseen raw DNA at inference time *without* region labels. We summarize GeneZip's empirical advantages along three axes:

• **Effective compression and prediction:** GeneZip's region-aware allocation generalizes from training sequences to held-out human chromosomes and an unseen species (mouse). GeneZip reaches 137.6 BPT with only a +0.31 perplexity increase. On downstream evaluation, GeneZip achieves the best average rank across four DNALongBench tasks—contact map prediction, eQTL prediction, enhancer–target gene prediction, and transcription-initiation signal prediction.

• **Redundancy-aware compression:** Our post-hoc RepeatMasker and TRF analyses show that GeneZip assigns stronger local compression to repetitive DNA sequence redundancy, including both TE-derived interspersed repeats and tandem-repeat (TR) interiors.

• **Efficient long-context modeling:** By reducing the effective token-mixing length, GeneZip enables 128K-context and 636M-parameter pretraining on a single A100 80GB GPU and fine-tunes on the DNALongBench eQTL task 50.4$\times$ faster than JanusDNA, making large-scale long-context DNA pretraining more practical.

2 Related Work

Dynamic chunking and adaptive tokenization. Most sequence models shorten raw inputs with a fixed tokenizer or a fixed pooling rule. H-Net reframes this step as end-to-end *dynamic chunking*: a learned router predicts content-dependent boundaries, and token mixing is performed on the resulting shorter sequence [10]. Its ratio loss supplies a global downsampling target without external segment labels. Recent DNA tokenization methods, including MxDNA, MergeDNA, VQDNA, and LDARNet, also move beyond fixed k-mers or BPE by learning sequence units, token merges, vector-quantized vocabularies, or adaptive boundaries [11–14]. These methods mainly address representation units or generic boundary learning. GeneZip addresses a different bottleneck: high-ratio, controllable compression for long-context DNA. Appendix G gives a direct comparison with DNA tokenization baselines; Section 4.2 audits the learned routing policy with post-hoc RepeatMasker/TRF masks; and Section 4.3 evaluates the downstream impact on DNALongBench. Concurrent H-Net-style genomic work is closest in mechanism [15]; GeneZip adds region-aware budget supervision and bounded routing, allowing the router to spend a limited token budget according to gene-structure priors during training while requiring no annotations at inference.

DNA foundation models and long-context genomic prediction. DNA foundation models have progressed from k-mer/BPE Transformer encoders such as DNABERT and DNABERT-2, through multi-species Nucleotide Transformer models, to long-context architectures such as HyenaDNA, Caduceus, JanusDNA, Evo, and Evo 2 [4–6, 16–20]. A complementary line of sequence-to-function models, including Enformer, AlphaGenome, and NTv3, uses hierarchical U-Net-like processing to predict regulatory tracks and variant effects over hundreds of kilobases to megabase context [1, 2, 21]. These models motivate efficient long-context modeling, but their downsampling schedules are architecture-defined rather than tied to genomic region structure or task-specific resolution needs.

Compression for long-range genomic modeling. Existing long-range genomic models mostly reduce length uniformly, through fixed k-mers, strided convolutions, pooling, or task-agnostic learned boundaries [1, 2, 10, 16]. Uniform allocation is mismatched to the non-uniform information density of genomes: coding exons occupy only $\sim$1–2% of the human genome and are enriched for evolutionary constraint [22, 23], while regulatory signal is distributed across promoters, UTRs, introns, enhancers, and distal intergenic regions in a task-dependent manner. GeneZip complements dynamic chunking with two mechanisms: region-aware ratio supervision, which turns gene-structure annotations into a controllable token-budget prior, and bounded routing, which prevents pathological over- or under-routing during large-scale training. Empirically, GeneZip achieves region-aware and redundancy-aware compression: Appendix C measures transfer of region-aware allocation to held-out

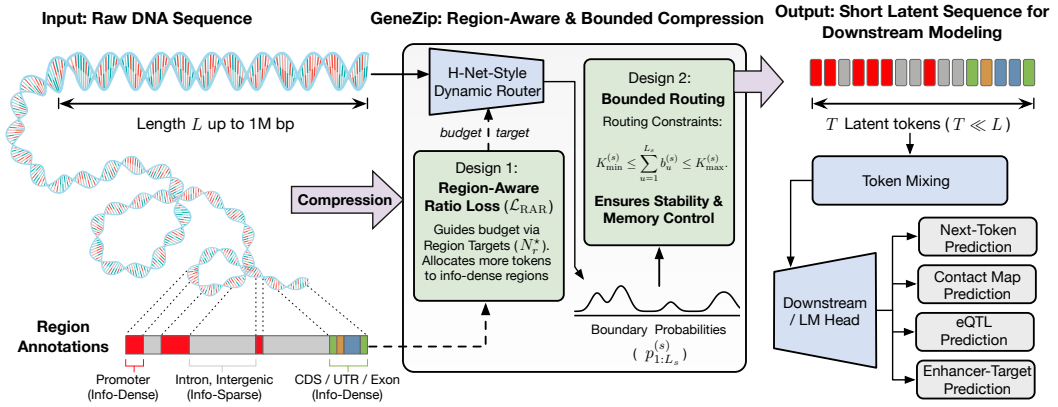


Figure 1: GeneZip overview. GeneZip compresses long-context DNA into a much shorter latent token sequence via hierarchical dynamic routing. During training, the Region-Aware Ratio (RAR) loss learns a region-adaptive token budget, and bounded routing enforces global token-count constraints for stable training. The compressed tokens are then processed by an arbitrary token-mixing backbone for downstream long-range tasks. After training, GeneZip compresses unseen raw DNA sequences *without* requiring annotations or repeat masks at inference time.

human chromosomes and mouse windows; Section 4.1 reports the reconstruction ability in terms of perplexity. Section 4.2 shows stronger local compression on predictable repeat subsets, especially LINE/SINE/TR cases.

3 Methodology

3.1 Problem setup and motivation

Let $x_{1:L}$ denote a DNA sequence of length L base pairs. Many long-range genomics tasks require contexts on the order of 10^5–10^6 bp. We decompose long-range DNA modeling into three stages—encoding, token mixing, and decoding:

$$\begin{aligned} z_{1:T} &= \mathrm{Encode}(x_{1:L}), \qquad T \ll L, \\ z'_{1:T} &= \mathrm{TokenMixing}(z_{1:T}), \\ \hat{y} &= \mathrm{Decode}(z'_{1:T}). \end{aligned} \tag{1}$$

Here $\mathrm{Encode}(\cdot)$ compresses raw DNA into a shorter latent sequence $z_{1:T}$, $\mathrm{TokenMixing}(\cdot)$ models interactions among the T latent tokens, and $\mathrm{Decode}(\cdot)$ maps mixed representations to pre-training or downstream task outputs.

In long-context DNA models, the dominant compute and memory typically sit in the token-mixing backbone (e.g., Transformer layers [24] or long-sequence state-space mixers [4, 5]), while encoding and decoding are comparatively lightweight. Models therefore downsample early so deep layers operate on a shorter sequence. Standard fixed-stride or pooling operators, however, allocate uniform resolution to all genomic positions [1, 2]. This is mismatched to genomic information density: promoters, splice junctions, UTR boundaries, coding regions, regulatory motifs, and distal regulatory compartments matter differently across tasks. GeneZip addresses this by realizing $\mathrm{Encode}(\cdot)$ as learned non-uniform compression over long-context DNA (Figure 1). H-Net-style dynamic routing proposes variable-length chunks [10]; the Region-Aware Ratio (RAR) loss supervises region-wise token budgets from gene-structure annotations (CDS, UTR, exon, intron, promoter, and intergenic regions). The compressor is trained without functional assay labels such as Hi-C, CAGE, ATAC/ChIP, or GTEx, and without RepeatMasker or TRF repeat labels. Throughout, we measure compression by base-pairs-per-token, $\mathrm{BPT} := L/T$.

3.2 GeneZip: hierarchical genomic compression with region-aware supervision

GeneZip instantiates $\mathrm{Encode}(\cdot)$ with a multi-stage hierarchical genomic compression mechanism inspired by H-Net [10]. At each stage, the encoder (i) adaptively partitions a long sequence into variable-length chunks and (ii) pools each chunk into one token, so that deep token mixing can be applied on a much shorter sequence. Unlike fixed-stride pooling, the chunk boundaries are learned and depend on the input sequence.

Dynamic Routing module. GeneZip maintains continuous sequence representations at multiple resolutions. We start from base-level embeddings $h^{(1)}_{1:L_1} = \mathrm{Embed}(x_{1:L})$ with $L_1 = L$. At routing stage $s \in \{1, \dots, S\}$, the stage-s encoder $E^{(s)}$ produces per-position representations $\hat{h}^{(s)}_{1:L_s} = E^{(s)}\left(h^{(s)}_{1:L_s}\right)$, where L_s is the current sequence length at stage s. Following H-Net [10], we induce adaptive boundaries by measuring the representation shift between adjacent positions in $\hat{h}^{(s)}$. We compute boundary probabilities $p^{(s)}_t$ via cosine *dissimilarity* between adjacent projected representations:

$$q^{(s)}_t = W^{(s)}_q \hat{h}^{(s)}_t, \qquad k^{(s)}_t = W^{(s)}_k \hat{h}^{(s)}_t, \tag{2}$$

$$p^{(s)}_t = \begin{cases} 1, & t = 1, \\ \frac{1}{2}\left(1 - \dfrac{\left(q^{(s)}_t\right)^\top k^{(s)}_{t-1}}{\left\|q^{(s)}_t\right\| \left\|k^{(s)}_{t-1}\right\|}\right), & t = 2, \dots, L_s, \end{cases} \tag{3}$$

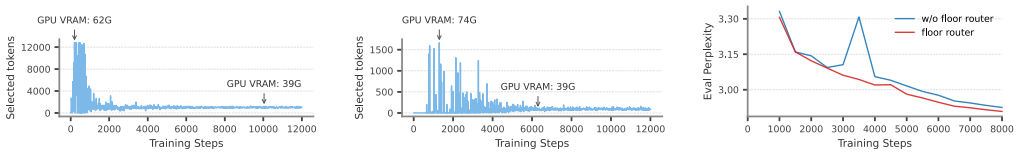
$$p^{(s)}_t \in [0, 1], \qquad \tilde{b}^{(s)}_t = \mathbf{1}\left\{p^{(s)}_t \geq 0.5\right\}. \tag{4}$$

By definition $p^{(s)}_1 = 1$, ensuring each sequence begins with a boundary. Intuitively, when consecutive representations span a contextual shift, their cosine similarity decreases and the corresponding boundary probability $p^{(s)}_t$ increases.

Hierarchical dynamic chunking. At routing stage $s \in \{1, \dots, S\}$, the encoder operates on a sequence of length L_s and predicts boundary probabilities $\{p^{(s)}_t\}_{t=1}^{L_s}$ using the routing module above. We first threshold them into a provisional boundary mask $\{\tilde{b}^{(s)}_t\}_{t=1}^{L_s}$, and then apply bounded routing (Sec. 3.3) to obtain a length-controlled final mask $b^{(s)} = \Pi(\tilde{b}^{(s)}; p^{(s)})$. The final mask $\{b^{(s)}_t\}_{t=1}^{L_s}$ partitions the sequence into variable-length chunks. Each chunk is pooled into a single representation, producing a shorter sequence of length $L_{s+1} = \sum_{t=1}^{L_s} b^{(s)}_t$. The pooled sequence is then served as the input for the further stage, denoted as $h^{(s+1)}_{1:L_{s+1}}$. After S stages, the encoder outputs $z_{1:T} = h^{(S+1)}_{1:L_{S+1}}$ with $T = L_{S+1}$. Compared to uniform striding, the learned boundary policy yields non-uniform resolution: segments with frequent boundaries receive more tokens.

Token selection, dechunking, and gradient estimator. We use the same downsampling and upsampling interface as H-Net [10]. In the downsampling path, the router's hard mask retains boundary-marked hidden states as chunk representatives; these retained states form the shorter sequence processed by the next compression stage and, after the final stage, by the token-mixing backbone. For base-level next-token pre-training, the mixed compressed sequence is decoded through the hierarchy in the reverse direction. Each compressed representation is expanded back to the positions covered by its corresponding chunk, combined with the encoder-side residual stream at that resolution, and refined by a lightweight decoder network. A nucleotide prediction head is then applied to the restored base-resolution representations. We also adopt the H-Net smoothing module, so uncertain boundaries interpolate neighboring chunk representations instead of relying on a purely hard assignment. The hard boundary mask is used in the forward pass, while the straight-through estimator and the smoothing path provide gradients to the boundary probabilities. As a result, the base-level next-token loss trains the encoder, router, token mixer, and decoder end-to-end.

Region supervision via gene structure. To make the compression region-aware and controllable, we supervise the *region-wise* token density using gene-structure annotations. Concretely, each base position t is assigned a genomic region label $r(t) \in \mathcal{R}$ (e.g. CDS, promoter, and intergenic regions). These region labels are used only to define supervision targets (RAR); the boundary predictor itself is trained to depend only on sequence, enabling annotation-free compression at inference. Repeat



(a) Selected tokens in stage 1 (b) Selected tokens in stage 2 (c) Eval PPL (with and without floor)

Figure 2: **Why bounded routing is necessary.** (a) In stage 1 compression, the router can be highly unstable early on and occasionally selects an excessively large number of tokens, which risks memory blow-up; a *ceiling* constraint prevents such pathological spikes. (b) In stage 2 compression, the router may collapse to selecting too few tokens (often near 1) at the beginning, which can induce training spikes and slow down optimization; a *floor* constraint enforces a minimum token budget for stable learning. (c) The floor constraint yields smoother and more favorable perplexity trajectories compared to training without it.

labels are never used as supervision; they are used only post hoc to test whether the learned router compresses repetitive DNA sequence redundancy.

We specify a base BPT anchor N and a region multiplier μ_r for each $r \in \mathcal{R}$, yielding a region target

$$N_r^\star = N \cdot \mu_r. \tag{5}$$

Intuitively, smaller μ_r means *higher* resolution (more tokens) and larger μ_r means *stronger* compression. For example, one can set multipliers such as: promoter $= 1$, CDS $= 1$, UTR $= 2$, exon $= 2$, intron $= 8$, near-gene intergenic $= 8$, distal-intergenic $= 16$, to encourage GeneZip to allocate more tokens to promoter and genic compartments. Other downstream tasks can select different multipliers when their validation signal favors regulatory, intergenic, or more balanced resolution.

Region-Aware Ratio (RAR) loss. RAR enforces region-specific compression by aligning region-wise keep rates across S routing stages. Given the region target $N_r^\star$ (Eq. 5), we geometrically factorize it over stages and define the per-stage target compression and keep rate:

$$N_r^{\star(s)} \triangleq (N_r^\star)^{1/S}, \qquad \tau_r^{\star(s)} \triangleq \frac{1}{N_r^{\star(s)}}, \qquad s = 1, \dots, S. \tag{6}$$

At stage s, each position $t \in \{1, \dots, L_s\}$ is associated with a region label $r^{(s)}(t) \in \mathcal{R}$ by propagating base-level annotations through the current segmentation (e.g., assigning each chunk the majority region label of the bases it covers). Let $L_r^{(s)} \triangleq \sum_{t=1}^{L_s} \mathbf{1}\{r^{(s)}(t) = r\}$. We define the empirical/expected keep rates:

$$\begin{aligned} F_r^{(s)} &\triangleq \frac{1}{L_r^{(s)}} \sum_{t=1}^{L_s} \mathbf{1}\{r^{(s)}(t) = r\}\, b_t^{(s)}, \\ G_r^{(s)} &\triangleq \frac{1}{L_r^{(s)}} \sum_{t=1}^{L_s} \mathbf{1}\{r^{(s)}(t) = r\}\, p_t^{(s)}. \end{aligned} \tag{7}$$

We then apply the ratio loss with target compression $N_r^{\star(s)}$:

$$\mathcal{L}_{\mathrm{RAR}}^{(r,s)} = \frac{N_r^{\star(s)}}{N_r^{\star(s)} - 1} \Big((N_r^{\star(s)} - 1)\, F_r^{(s)}\, G_r^{(s)} + (1 - F_r^{(s)})(1 - G_r^{(s)}) \Big). \tag{8}$$

Intuition. RAR couples the realized keep rate $F_r^{(s)}$ after projection with its expectation $G_r^{(s)}$ before projection to match $\tau_r^{\star(s)}$. When $F_r^{(s)} = G_r^{(s)}$, Eq. (8) is minimized at $F_r^{(s)} = G_r^{(s)} = \tau_r^{\star(s)}$ with $\mathcal{L}_{\mathrm{RAR}}^{(r,s)} = 1$; if $F_r^{(s)}$ is too high/low, gradients push $p_t^{(s)}$ down/up in region r. Aggregating by region mass gives:

$$\mathcal{L}_{\mathrm{RAR}} = \sum_{s=1}^{S} \sum_{r \in \mathcal{R}} \pi_r^{(s)}\, \mathcal{L}_{\mathrm{RAR}}^{(r,s)}, \qquad \pi_r^{(s)} \triangleq \frac{L_r^{(s)}}{L_s}. \tag{9}$$

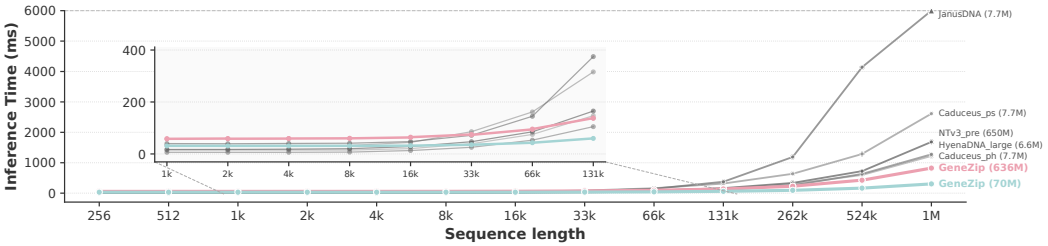


Figure 3: **Inference efficiency on long-context inputs.** End-to-end inference latency as a function of input sequence length (256 bp to 1M bp). GeneZip (70M/636M) remains consistently fast across the full range, while baseline long-context models exhibit sharply increasing latency as sequence length grows. The inset zooms into the short-to-mid regime (1K–131K) to highlight differences at smaller contexts.

3.3 Bounded Routing for Stable Training

Under aggressive compression (Figure 2), unconstrained routing can select too many boundaries, causing GPU memory spikes, or too few, causing capacity collapse. We stabilize training by projecting routing decisions onto a per-stage token-budget interval.

Reference token budget. At routing stage $s \in \{1, \dots, S\}$, the router predicts boundary probabilities $p^{(s)}_{1:L_s}$ and produces a provisional hard mask $\tilde{b}^{(s)}$ (e.g., by thresholding). We use a stage-wise keep-rate target $\tau^{(s)}$ as the reference budget, where $\prod_{s=1}^{S} \tau^{(s)} = 1/N$ and we use the uniform schedule $\tau^{(s)} = N^{-1/S}$ in this work. Given current length L_s, the reference token budget is $K^{(s)} \triangleq \tau^{(s)} L_s$. We define a floor and ceiling around $K^{(s)}$:

$$K^{(s)}_{\min} = \rho^{(s)}_{\min} K^{(s)}, \qquad K^{(s)}_{\max} = \rho^{(s)}_{\max} K^{(s)}. \tag{10}$$

Projection (bounded routing). We compute a projected mask $b^{(s)} = \Pi(\tilde{b}^{(s)}; p^{(s)})$ that satisfies

$$K^{(s)}_{\min} \le \sum_{t=1}^{L_s} b^{(s)}_t \le K^{(s)}_{\max}. \tag{11}$$

The projection flips the fewest boundary decisions using $p^{(s)}$ as confidence: if too few boundaries are selected, it activates the largest-$p^{(s)}_t$ inactive positions; if too many are selected, it deactivates the smallest-$p^{(s)}_t$ active positions. The first position in each segment is always kept.

3.4 Training and Inference

Training objective. Following H-Net [10], we train GeneZip end-to-end with a language-modeling objective combined with the multi-stage RAR regularizer. Let $\mathcal{L}_{\text{NTP}}$ denote base-level next-token cross-entropy. The overall objective is: $\mathcal{L} = \mathcal{L}_{\text{NTP}} + \alpha \mathcal{L}_{\text{RAR}}$, where α controls the strength of region-aware compression supervision.

Annotation-free inference. At inference, GeneZip compresses unseen DNA *without* annotations: sequence-predicted boundaries and bounded routing produce tokens that can be fed into any token-mixing backbone (Transformer/Mamba) and decoded for downstream tasks.

Inference efficiency and long-context scaling. Figure 3 reports end-to-end inference latency as a function of input length up to 1M bp (benchmarking details in Appendix A.2). Across all tested contexts, GeneZip remains consistently fast, and the latency gap widens as sequences enter the hundreds-of-kilobases to megabase regime. Baseline long-context models that operate closer to base-level resolution show rapidly increasing runtime with length and multi-second latency at the longest inputs. GeneZip reduces the *effective* token-mixing length via region-adaptive compression, so computation is dominated by mixing over a much shorter latent sequence. This scaling profile supports GeneZip as a practical tokenizer-level infrastructure for long-context DNA modeling and complements the end-to-end training-time gains reported in Table 4.

Table 1: Region-wise autoregressive perplexity (PPL) on the validation dataset. We omit `dig` (distal-intergenic) because it is weakly constrained and often heavily repeated, making PPL less informative. **Bold** indicates the best (lowest) PPL among encoder-compression methods (U-Net, H-Net, GeneZip). Underline indicates the best PPL of small model trained at 12.8K length.

Model	BPT	Overall ↓	Prom. ↓	CDS ↓	UTR ↓	Exon ↓	Intron ↓	NIG ↓
Backbone / fine-grained upper bounds								
Mamba2-84M	1.0	2.4135	2.0494	2.4136	2.5608	2.6346	2.5076	2.4826
H-Net-3BPT-70M	3.0	2.4115	2.0694	2.4815	2.4306	2.8113	2.5462	2.4357
Uniform / hierarchical compression baselines								
U-Net-70M	128.0	2.8414	3.1335	3.4665	3.2389	3.0586	2.8415	2.6474
H-Net-128BPT-70M	122.2	2.7930	2.8940	3.4657	3.0241	3.0277	2.8270	2.6375
GeneZip (region-aware compression)								
GeneZip-70M	137.6	2.7259	2.6765	3.4124	**2.9084**	2.9847	2.7761	2.6278
GeneZip-70M-128K	169.2	2.6557	2.7520	3.0866	3.1236	3.0401	2.7350	2.5422
GeneZip-636M	137.0	2.6620	**2.6094**	**2.8272**	2.9311	3.2052	2.7724	2.5741
GeneZip-636M-128K	161.1	**2.6484**	2.8121	2.9647	3.1037	**2.8712**	**2.6486**	**2.4941**

4 Experiments

The experiments are organized around three claims. First, GeneZip is effective: Section 4.1 evaluates pretraining PPL and BPT under aggressive compression, Appendix C tests whether the learned region-allocation policy transfers to unseen human chromosomes and an unseen species (mouse) without inference-time annotations, and Appendix D visualizes region-aligned routing at representative loci. Second, GeneZip is redundancy-aware: Section 4.2 tests whether it preferentially compresses repetitive DNA sequence redundancy—both TE-derived interspersed repeats and tandem-repeat interiors—without repeat labels. Third, GeneZip is efficient and useful for downstream long-context tasks: the pretraining study includes 128K-context and 636M-parameter GeneZip variants, the eQTL benchmark shows a 50.4× wall-clock fine-tuning speedup over JanusDNA, and four reproducible DNALongBench tasks—CMP, eQTL, ETGP, and TISP—show the best average rank among all compared sequence models and the strongest leakage-controlled results on CMP, eQTL, and ETGP (Section 4.3 and Appendix B).

4.1 Pretraining analysis: compression quality and scaling

We first evaluate GeneZip's language modeling performance and compare it with Mamba2 [25], U-Net [26], and H-Net [10]. For U-Net, we implement an autoregressive U-Net variant of our model by replacing the encoder with an autoregressive U-Net. For H-Net, we train two variants: a low-compression setting (H-Net-3BPT, following the default configuration in [10]) and a high-compression setting (H-Net-128BPT). We train comparable-size baselines (70M-84M) and GeneZip on 100B base pairs of GENCODE data using a 12.8K context length. For GeneZip, we additionally study length scaling by continuing pretraining with a 128K context and also evaluate a 636M-parameter model. We report pretraining performance as per-region perplexity (PPL) on the validation set. For a detailed compression analysis, please refer to Appendix C.

Table 1 reports the overall evaluation PPL along with region-stratified PPL. We observe the following trends. First, Mamba2 and the low-compressed H-Net-3BPT serve as strong reference points, achieving the lowest PPLs under fixed or very fine-grained tokenizations. As expected, increasing the compression rate generally degrades performance; for example, H-Net-128BPT incurs a +0.38 PPL increase relative to the low-compression H-Net setting. Second, among high-compression methods, GeneZip variants achieve the best overall PPL while operating at substantially higher effective compression. For instance, among 12.8K-context 70M compressors, GeneZip-70M is the strongest highly compressed model: it uses the strongest compression level ($\mathrm{BPT} = 137.6$) and still attains the lowest PPL across all reported regions. This suggests that GeneZip learns to identify region structure and allocate tokens more effectively. Finally, GeneZip exhibits favorable scaling with both context length and model size. Continuing pretraining to 128K context improves performance: GeneZip-70M-128K outperforms its 12.8K counterpart (GeneZip-70M) and incurs only a +0.24 PPL increase despite a more aggressive compression level ($\mathrm{BPT} = 169.2$). The best results are obtained

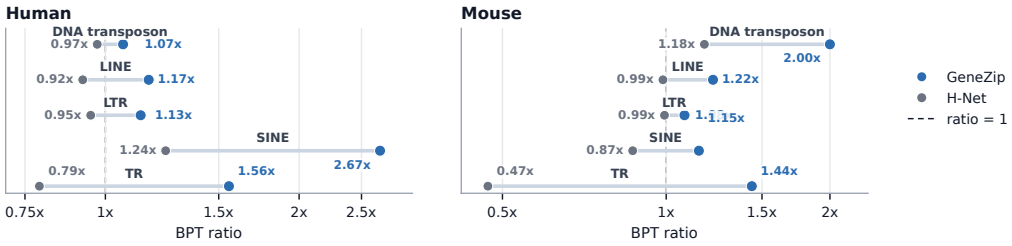


Figure 4: **Redundancy-aware compression across TE-derived and tandem-repeat sequence.** Repeat positions are compared with matched non-repeat positions in the same full-window context. BPT ratio above 1 means stronger local compression; PPL details are in Table 9. GeneZip converts repeat predictability into higher local compression, especially for LINE, human SINE, and TR, whereas H-Net often assigns equal or lower BPT to the same repetitive subsets.

by GeneZip-636M-128K, indicating clear gains from combining longer-context pretraining with increased model capacity.

4.2 Redundancy-aware compression of repetitive DNA

Repetitive DNA is a major source of sequence-level redundancy in mammalian genomes. We audit two complementary post-hoc masks: RepeatMasker TE classes (DNA transposon, LINE, LTR, and SINE), which cover interspersed TE-derived repeats, and TRF tandem repeats (TR), which capture local periodicity [9, 27]. Neither mask is used in GeneZip or H-Net training; they are used only to test whether raw-sequence compression spends fewer tokens on predictable repeat interiors. We evaluate redundancy-aware behavior on two matched 10 Mb samples: 78 non-overlapping 128K held-out human windows from `chr8/chr9/chr10`, and 78 mouse evaluation windows. RepeatMasker transposable-element (TE) masks and Tandem Repeats Finder (TRF) tandem-repeat (TR) masks are used only post hoc. For each repeat subset, we compare repeat positions with matched non-repeat positions within the same full-window context and report repeat/non-repeat ratios for both PPL and BPT. A PPL ratio below 1 indicates easier reconstruction, whereas a BPT ratio above 1 indicates stronger local compression.

Table 9 first provides a predictability check: most TE/TR subsets have lower repeat-position PPL than their matched backgrounds, indicating that repeat interiors are generally easier to reconstruct. This motivates using TE/TRs as redundancy probes, but predictability alone does not imply redundancy-aware routing. Figure 4 shows the corresponding compression behavior. GeneZip converts repeat predictability into more aggressive local compression: on human windows, its BPT ratios are 2.67 for SINE, 1.56 for TR, and 1.17 for LINE; on mouse windows, they are 1.44 for TR, 1.22 for LINE, and 1.15 for SINE. H-Net does not exhibit the same allocation pattern: its BPT ratios stay near or below 1 for most TE classes, and it compresses TR less than the matched background (0.79 in human and 0.47 in mouse). Thus, without TE/TR supervision, GeneZip learns to spend fewer tokens on predictable repeat interiors. Together, these results support **redundancy-aware compression**: GeneZip identifies repeat-associated redundancy and compresses it more aggressively than matched non-repeat sequence, whereas H-Net shows weaker and less consistent repeat-specific compression.

4.3 DNALongBench downstream evaluation

We evaluate downstream transfer on four reproducible DNALongBench [3] tasks: contact map prediction (CMP), eQTL prediction, enhancer-target gene prediction (ETGP), and transcription initiation signal prediction (TISP). We organize the baselines into three groups. *DNALongBench-official* contains the official DNALongBench results for HyenaDNA, Caduceus-Ph, and Caduceus-PS. *External-pretraining* contains JanusDNA, NTv3-100M-pre, and Evo2-1B; these models use pretraining data different from GeneZip's GENCODE corpus, so their pretraining may overlap benchmark regions or related functional sources, but we evaluate them with the same downstream fine-tuning and scoring pipeline where rerun. *Leakage-controlled-pretraining* contains Mamba2, U-Net, H-Net-128BPT, and GeneZip-70M, all trained on the same GENCODE-12.8K corpus for 100B base pairs, with DNALongBench validation/test intervals and chromosomes `chr8/chr9/chr10` excluded, and evaluated with matched downstream fine-tuning.

Table 2: All-task summary on DNALongBench [3]. Time column reports the fine-tuning time of language models measured on the eQTL *Artery_Tibial* task. GeneZip-70M reports task-wise validation-selected RAR priors, with the fixed transcript-balanced prior reported in Table 10. Avg. rank is computed over the four task metrics, with lower being better. Best task-metric results within *Leakage-controlled-pretraining* are in **bold**; best results across all rows are underlined.

Model	Time (min)	CMP SCC	eQTL AUROC	ETGP AUPRC	TISP PCC	Avg. rank
DNALongBench-official						
HyenaDNA[4]	210	0.115	0.514	0.249	0.132	7.5
Caduceus-Ph[5]	435	0.133	0.566	0.380	0.109	5.5
Caduceus-PS[5]	891	0.127	0.538	0.376	0.108	6.8
External-pretraining						
JanusDNA[6]	2520	0.159	0.840	0.438	0.002	3.5
NTv3-100M-pre[21]	117	0.136	0.792	0.440	0.194	3.0
Evo2-1B[20]	1304	0.018	0.754	0.095	-0.002	9.3
Leakage-controlled-pretraining						
Mamba2[25]	328	0.111	0.762	0.322	**0.294**	5.5
U-Net[26]	39	0.131	0.769	0.244	0.008	6.5
H-Net-128BPT[10]	65	0.118	0.798	0.131	0.235	5.5
GeneZip-70M	50	**0.133**	**0.820**	**0.446**	0.279	**2.0**

Table 2 summarizes one task-level metric per benchmark (full results in Appendix B), together with fine-tuning time measured on the eQTL *Artery_Tibial* task. Compression first improves downstream efficiency: within the leakage-controlled-pretraining group, U-Net, H-Net-128BPT, and GeneZip reduce fine-tuning time from 328 minutes for the uncompressed Mamba2 backbone to 39, 65, and 50 minutes, respectively. This speedup, however, often comes with an accuracy cost. H-Net-128BPT trails Mamba2 on CMP, ETGP, and TISP, while U-Net drops sharply on ETGP and TISP despite competitive CMP and eQTL scores. GeneZip largely avoids this degradation: it is best within this group on CMP, eQTL, and ETGP, and only trails Mamba2 on TISP. Across all rows, GeneZip achieves the best average rank. Thus, even with $> 100\times$ compression, GeneZip delivers the strongest overall downstream performance across this long-context suite.

GeneZip is also flexible across downstream tasks because different tasks depend on different genomic compartments. TISP is promoter-centered; eQTL benefits from broader cis-regulatory context across promoter, UTR, intronic, and near-gene intergenic regions; ETGP depends on intergenic/noncoding regulatory context for enhancer–target links; and CMP integrates 1-Mb local folding signals, including CTCF-associated loops, TAD/sub-TAD structure, enhancer–promoter interactions, and local compartment-like boundary signals. Uniform compression assigns the same resolution target everywhere and cannot adapt token budgets to region-specific information density. GeneZip instead exposes regional resolution as a controllable allocation interface: by setting the region multipliers μ_r, it can preserve the compartments expected to carry the downstream signal while compressing less informative regions more aggressively. The μ_r ablations are reported in Appendix E.1.

5 Conclusion

We propose GeneZip, a region-aware DNA compressor that combines bounded dynamic routing with a region-aware compression objective to reallocate token budget across genomic compartments. GeneZip achieves $137.6\times$ compression with only a 0.31 perplexity increase, preserves downstream utility across long-context genomic tasks, and obtains the best aggregate rank across four reproducible DNALongBench tasks among all compared sequence models. A post-hoc repeat analysis shows redundancy-aware behavior: without TE or TR supervision, GeneZip assigns higher local compression to repetitive DNA sequence redundancy than generic H-Net routing, with robust cross-species signals for LINE and tandem repeats and a particularly sharp human SINE effect. GeneZip also supports 128K-context and 636M-parameter training within the reported single-A100 setup and substantially reduces wall-clock cost for downstream long-context fine-tuning. These results establish GeneZip as an effective, redundancy-aware, and efficient compression interface for long-context DNA modeling.

References

[1] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwińska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, 2021. doi: 10.1038/s41592-021-01252-x.

[2] 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(8099):1206–1218, January 2026. doi: 10.1038/s41586-025-10014-0.

[3] Wenduo Cheng, Zhenqiao Song, Yang Zhang, Shike Wang, Danqing Wang, Muyu Yang, Lei Li, and Jian Ma. DNALONGBENCH: a benchmark suite for long-range DNA prediction tasks. *Nature Communications*, 16(1):10108, 2025. doi: 10.1038/s41467-025-65077-4.

[4] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Christopher Ré. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution, June 2023. NeurIPS 2023 Spotlight.

[5] 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*, pages 43632–43648. PMLR, 2024.

[6] Qihao Duan, Bingding Huang, Zhenqiao Song, Irina Lehmann, Lei Gu, Roland Eils, and Benjamin Wild. JanusDNA: A powerful bi-directional hybrid DNA foundation model, May 2025. NeurIPS 2025.

[7] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. doi: 10.1038/nature11247. URL `https://www.nature.com/articles/nature11247`.

[8] Adam Frankish, Mark Diekhans, Anne-Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M. Mudge, Cristina Sisu, James Wright, Joel Armstrong, Ian F. Barnes, Andrew Berry, Alexandra Bignell, Silvia Carbonell Sala, Jacqueline Chrast, Fiona Cunningham, Tomás Di Domenico, Sarah Donaldson, Ian T. Fiddes, Carlos García Girón, Jose Manuel Gonzalez, Tiago Grego, Matthew Hardy, Thibaut Hourlier, Toby Hunt, Osagie G. Izuogu, Julien Lagarde, Fergal J. Martin, Laura Martínez, Shamika Mohanan, Paul Muir, Fabio C. P. Navarro, Anne Parker, Baikang Pei, Fernando Pozo, Magali Ruffier, Bianca M. Schmitt, Eloise Stapleton, Marie-Marthe Suner, Irina Sycheva, Barbara Uszczynska-Ratajczak, Jinuri Xu, Andrew Yates, Daniel Zerbino, Yan Zhang, Bronwen Aken, Jyoti S. Choudhary, Mark Gerstein, Roderic Guigó, Tim J. P. Hubbard, Manolis Kellis, Benedict Paten, Alexandre Reymond, Michael L. Tress, and Paul Flicek. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Research*, 47(D1):D766–D773, 2019. doi: 10.1093/nar/gky955.

[9] Guillaume Bourque, Kathleen H. Burns, Mary Gehring, Vera Gorbunova, Andrei Seluanov, Molly Hammell, Michaël Imbeault, Zsuzsanna Izsvák, Henry L. Levin, Todd S. Macfarlan, Dixie L. Mager, and Cédric Feschotte. Ten things you should know about transposable elements. *Genome Biology*, 19(1):199, 2018. doi: 10.1186/s13059-018-1577-z. URL `https://doi.org/10.1186/s13059-018-1577-z`.

[10] Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling, July 2025. ICLR 2026 Poster.

[11] 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, December 2024. NeurIPS 2024.

[12] Siyuan Li, Kai Yu, Anna Wang, Zicheng Liu, Chang Yu, Jingbo Zhou, Qirong Yang, Yucheng Guo, Xiaoming Zhang, and Stan Z. Li. MergeDNA: Context-aware genome modeling with dynamic tokenization through token merging, November 2025. AAAI 2026 Oral Presentation Preprint.

[13] 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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28717–28733. PMLR, July 2024. URL `https://proceedings.mlr.press/v235/li24bm.html`.

[14] Daria Ledneva and Denis Kuznetsov. LDARNet: DNA adaptive representation network with learnable tokenization for genomic modeling. OpenReview, ICLR 2026 Conference Submission, September 2025. Under review.

[15] Arnav Shah, Junzhe Li, Parsa Idehpour, Adibvafa Fallahpour, Brandon Wang, Sukjun Hwang, Bo Wang, Patrick D. Hsu, Hani Goodarzi, and Albert Gu. dnaHNet: A scalable and hierarchical foundation model for genomic sequence learning, February 2026.

[16] 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.

[17] 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 *International Conference on Learning Representations*, 2024.

[18] 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.

[19] 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, Jeremy Sullivan, Madelena Y Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D Hsu, and Brian L Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336. URL `https://pubmed.ncbi.nlm.nih.gov/39541441/`.

[20] Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Jonathan C. Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, 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.

[21] 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, Alexander 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, December 2025. URL `https://www.biorxiv.org/content/10.64898/2025.12.22.695963v1`. bioRxiv preprint.

[22] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. doi: 10.1038/35057062. URL `https://www.nature.com/articles/35057062`.

[23] Kerstin Lindblad-Toh, Manuel Garber, Or Zuk, Michael F. Lin, Brian J. Parker, Stefan Washietl, Pouya Kheradpour, Jason Ernst, Gregory Jordan, Evan Mauceli, Lucas D. Ward, Craig B. Lowe, Alisha K. Holloway, Michele Clamp, Sante Gnerre, Jessica Alföldi, Kathryn Beal, Jean Chang, Hiram Clawson, James Cuff, Federica Di Palma, Stephen Fitzgerald, Paul Flicek, Mitchell Guttman, Melissa J. Hubisz, David B. Jaffe, Irwin Jungreis, W. James Kent, Dennis Kostka, Marcia Lara, Andre L. Martins, Tim Massingham, Ida Moltke, Brian J. Raney, Matthew D. Rasmussen, Jim Robinson, Alexander Stark, Albert J. Vilella, Jiayu Wen, Xiaohui Xie, Michael C. Zody, Broad Institute Sequencing Platform and Whole Genome Assembly Team, Jen Baldwin, Toby Bloom, Chee Whye Chin, Dave Heiman, Robert Nicol, Chad Nusbaum, Sarah Young, Jane Wilkinson, Kim C. Worley, Christie L. Kovar, Donna M. Muzny, Richard A. Gibbs, Baylor College of Medicine Human Genome Sequencing Center Sequencing Team, Andrew Cree, Huyen H. Dihn, Gerald Fowler, Shalili Jhangiani, Vandita Joshi, Sandra Lee, Lora R. Lewis, Lynne V. Nazareth, Geoffrey Okwuonu, Jireh Santibanez, Wesley C. Warren, Elaine R. Mardis, George M. Weinstock, Richard K. Wilson, Genome Institute at Washington University, Kim Delehaunty, David Dooling, Catrina Fronik, Lucinda Fulton, Bob Fulton, Tina Graves, Patrick Minx, Erica Sodergren, Ewan Birney, Elliott H. Margulies, Javier Herrero, Eric D. Green, David Haussler, Adam Siepel, Nick Goldman, Katherine S. Pollard, Jakob S. Pedersen, Eric S. Lander, and Manolis Kellis. A high-resolution map of human evolutionary constraint using 29 mammals. *Nature*, 478(7370):476–482, 2011. doi: 10.1038/nature10530.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[25] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10041–10071. PMLR, July 2024. URL `https://proceedings.mlr.press/v235/dao24a.html`.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28. URL `https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28`.

[27] Gary Benson. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Research*, 27(2):573–580, 1999. doi: 10.1093/nar/27.2.573. URL `https://doi.org/10.1093/nar/27.2.573`.

[28] Jian Zhou. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. *Nature Genetics*, 54(5):725–734, 2022. doi: 10.1038/s41588-022-01065-4.

[29] Alistair R. R. Forrest, Hideya Kawaji, Michael Rehli, J. Kenneth Baillie, Michiel J. L. de Hoon, Vanja Haberle, Timo Lassmann, Ivan V. Kulakovskiy, Marina Lizio, Masayoshi Itoh, Robin Andersson, et al. A promoter-level mammalian expression atlas. *Nature*, 507(7493):462–470, 2014. doi: 10.1038/nature13182.

[30] Robin Andersson, Claudia Gebhard, Irene Miguel-Escalada, Ilka Hoof, Jette Bornholdt, Mette Boyd, Yun Chen, Xiaobei Zhao, Christian Schmidl, Takeya Suzuki, Evgenia Ntini, et al. An atlas of active enhancers across human cell types and tissues. *Nature*, 507(7493):455–461, 2014. doi: 10.1038/nature12787.

[31] GTEx Consortium. The GTEx consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020. doi: 10.1126/science.aaz1776.

[32] Matthew T. Maurano, Richard Humbert, Eric Rynes, Robert E. Thurman, Eric Haugen, Hao Wang, Alex P. Reynolds, Richard Sandstrom, Hongzhu Qu, Jennifer Brody, Anthony Shafer, et al. Systematic localization of common disease-associated variation in regulatory DNA. *Science*, 337(6099):1190–1195, 2012. doi: 10.1126/science.1222794.

[33] Charles P. Fulco, Joseph Nasser, Thouis R. Jones, Glen Munson, Drew T. Bergman, Vidya Subramanian, Sharon R. Grossman, Rockwell Anyoha, Benjamin R. Doughty, Tejal A. Patwardhan, Tung H. Nguyen, et al. Activity-by-contact model of enhancer–promoter regulation from thousands of CRISPR perturbations. *Nature Genetics*, 51(12):1664–1669, 2019. doi: 10.1038/s41588-019-0538-0.

[34] Joseph Nasser, Drew T. Bergman, Charles P. Fulco, Philine Guckelberger, Benjamin R. Doughty, Tejal A. Patwardhan, Thouis R. Jones, Tung H. Nguyen, Jacob C. Ulirsch, Fritz Lekschas, Kristy Mualim, et al. Genome-wide enhancer maps link risk variants to disease genes. *Nature*, 593 (7858):238–243, 2021. doi: 10.1038/s41586-021-03446-x.

[35] Boris Lenhard, Albin Sandelin, and Piero Carninci. Metazoan promoters: emerging characteristics and insights into transcriptional regulation. *Nature Reviews Genetics*, 13(4):233–245, 2012. doi: 10.1038/nrg3163.

[36] Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009. doi: 10.1126/science.1181369.

[37] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012. doi: 10.1038/nature11082.

[38] Suhas S. P. Rao, Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, and Erez Lieberman Aiden. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014. doi: 10.1016/j.cell.2014.11.021.

[39] Geoff Fudenberg, David R Kelley, and Katherine S Pollard. Predicting 3D genome folding from DNA sequence with akita. *Nature Methods*, 17(11):1111–1117, 2020. doi: 10.1038/s41592-020-0958-x.

[40] Abdul Muntakim Rafi, Brett Kiyota, Nozomu Yachie, and Carl G. de Boer. Detecting and avoiding homology-based data leakage in genome-trained sequence models. *bioRxiv*, January 2025. doi: 10.1101/2025.01.22.634321. URL `https://www.biorxiv.org/content/10.1101/2025.01.22.634321v1`. Preprint.

[41] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, August 2016. doi: 10.18653/v1/P16-1162. URL `https://aclanthology.org/P16-1162/`.

A Implementation Details

A.1 Pretraining Settings

We pretrain on GENCODE-derived human corpora at 12.8K and 128K bp context lengths. Each window stores the raw sequence together with a run-length encoded segmentation over the 7-region taxonomy used throughout the paper (promoter, CDS, UTR, exon, intron, near-intergenic (NIG), and distal-intergenic (DIG)). To reduce DNALongBench leakage, we exclude benchmark validation/test intervals from `train/valid` sampling and hold out `chr8`, `chr9`, and `chr10`; full construction details and corpus statistics are provided in Appendix F and Table 12.

We use a two-stage hierarchical dynamic chunking encoder with $S = 2$, where each stage is implemented with two Mamba2 layers. The token-mixing backbone consists of 15 Mamba2 layers. We pretrain GeneZip with region-aware compression using this 7-region taxonomy. Across both stages, we fix the base BPT anchor to $N = 32$ bp/token. The default GeneZip model uses the transcript-balanced region-aware ratio targets over the seven regions (promoter:CDS:UTR:exon:intron:NIG:DIG $= 1 : 1 : 2 : 2 : 8 : 8 : 16$); Appendix E.1 studies alternative μ_r settings for task-aware allocation. For bounded routing, we apply a shared absolute floor across all GeneZip models with $K_{\min}^{(s)} = 8$ for all stages s, meaning at least the top-8 tokens are kept in each routing layer. The 70M model fits comfortably on an 80GB A100 and does not require a maximum constraint. The 636M model does require an upper bound, because unconstrained routing can trigger CUDA out-of-memory (OOM) errors. We therefore set $K_{\max}^{(1)} = 30{,}000$ and $K_{\max}^{(2)} = 10{,}000$ to prevent OOM. Unless otherwise noted, all stages of GeneZip models and baseline models are trained on a total budget of 100B base pairs each. All reported experiments except Evo2-1B can be run on a single NVIDIA A100 80GB or H100 GPU; Evo2-1B requires multi-GPU parallelism because of its model size and sharding constraints.

Phase 1 (12.8K) pretraining. We train on `gencode_human_12.8k` with a maximum sequence length of $12{,}800$ bp. We optimize with AdamW using learning rate 1×10^{-3}, weight decay 0.1, $(\beta_1, \beta_2) = (0.9, 0.95)$, a linear learning-rate schedule with 500 warmup steps, and gradient clipping to $\|g\|_2 \leq 2.0$. We use per-device batch size 32 with gradient accumulation 8 (effective batch size 256) and train for 100B base pairs using BF16 mixed precision. We evaluate using $3{,}000$ validation samples.

Phase 2 (128K) pretraining. We continue training on `gencode_human_128k` with a maximum sequence length of $128{,}000$ bp, initializing from the 12.8K checkpoint introduced above. We keep the same optimizer and schedule (AdamW; LR 1×10^{-3}; 500 warmup steps; weight decay 0.1; $(\beta_1, \beta_2) = (0.9, 0.95)$; gradient clipping 2.0; BF16). We use per-device batch size 4 with gradient accumulation 8 (effective batch size 32) and train for another 100B base pairs using BF16 mixed precision. We evaluate using $3{,}000$ validation samples.

A.2 Inference Latency Benchmark (Figure 3)

Figure 3 reports the inference-time scaling of each model as a function of input length. We measure the forward-pass latency on a single NVIDIA A100 80GB GPU with BF16 inference and batch size 1. For each context length $L \in \{256, 512, \ldots, 1{,}048{,}576\}$ bp, we construct a length-L DNA string by repeating a fixed sequence snippet and encode it into single-nucleotide token IDs.

Timing protocol. We time only the model forward pass, excluding model loading and input construction. For each (model, L) pair, we run 2 warm-up forward passes, followed by 10 timed runs. Each timed run synchronizes the device before and after the forward pass to record wall-clock time. We report the mean of the per-forward latency in milliseconds in Figure 3.

A.3 Existing assets and licenses/terms.

We cite the original sources for all third-party datasets, codebases, and public checkpoints used in the experiments. GENCODE v49 and the mouse GENCODE-derived annotations are open-access GENCODE project data.[1] The DNALongBench task data used here (CMP, eQTL, ETGP, and TISP)

[1] `https://www.gencodegenes.org/pages/data_access.html`

Table 3: Pre-training hyperparameters for the GeneZip models.

	12.8K pre-train	128K pre-train
Dataset	`gencode_human_12.8k`	`gencode_human_128k`
Region loss weight α	0.03	0.03
Max length (bp)	12,800	128,000
Per-device batch size	32	4
Grad. accumulation	8	8
Update steps	30,600	24,414
Optimizer	AdamW	AdamW
Learning rate	1×10^{-3}	1×10^{-3}
LR schedule	Linear + warmup	Linear + warmup
Warmup steps	500	500
Weight decay	0.1	0.1
Adam (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$
Grad clip	2.0	2.0
Precision	BF16	BF16
Validation samples	3,000	3,000

are distributed through Harvard Dataverse under CC0 1.0, and the DNALongBench code is released under BSD-3-Clause.[2] The public HyenaDNA, Caduceus, JanusDNA, and Evo2 code repositories are released under Apache-2.0 where applicable; their checkpoints, and the Nucleotide Transformer resources, are used under the corresponding public model-card or repository terms (Nucleotide Transformer: CC BY-NC-SA 4.0). We use these third-party assets only for research benchmarking and do not redistribute the original DNALongBench datasets or public baseline checkpoints in this work.

B Complete Results for DNALongBench

B.1 Task Selection

We attempted to include all tasks in DNALongBench. The regulatory sequence activity prediction (RSAP) release was not sufficiently reproducible for our pipeline. Specifically, the DNALongBench paper and official README describe RSAP as a 196,608-bp, 128-bp-bin task with 38,171 human and 33,521 mouse examples, with mouse split counts of 29,295/2,209/2,017 for train/validation/test and per-sample PCC over positions and tracks [3].[3] The released RSAP `statistics.json` files in the official Dataverse package report `seq_length=131072`, not 196,608, for both human and mouse.[4] The same official Dataverse file list also exposes an incomplete mouse TFRecord set: the visible mouse shards are `train-1-*` $\times 84$, `valid-1-{1,2,3,4,5,6,8}.tfr`, and `test-1-{0,1,2,5,6,7}.tfr`, with missing `valid-1-0.tfr`, `valid-1-7.tfr`, `test-1-3.tfr`, and `test-1-4.tfr`; this corresponds to 21,504/1,697/1,505 mouse records, below the paper's 29,295/2,209/2,017 split.[5] Finally, the released CNN configuration uses `nn.MSELoss()` for RSAP, while the paper states that RSAP CNNs were trained with Poisson loss.[6] We therefore report other tasks, i.e. CMP, eQTL, TISP, and ETGP, where the released data and scoring protocol could be validated under our pipeline.

[2] `https://github.com/ma-compbio/DNALONGBENCH`; task data DOIs are listed in the official README.

[3] Official paper: `https://www.nature.com/articles/s41467-025-65077-4`; official README: `https://github.com/ma-compbio/DNALONGBENCH`.

[4] Human statistics: `https://dataverse.harvard.edu/api/access/datafile/10443886`; mouse statistics: `https://dataverse.harvard.edu/api/access/datafile/10443890`.

[5] RSAP Dataverse dataset: `https://doi.org/10.7910/DVN/MNUEZR`; metadata API: `https://dataverse.harvard.edu/api/datasets/:persistentId/?persistentId=doi:10.7910/DVN/MNUEZR`.

[6] Official CNN config: `https://github.com/ma-compbio/DNALONGBENCH/blob/main/experiments/CNN/task_configs.py`.

Table 4: DNALongBench eQTL tasks (AUROC; higher is better). The best result in each dataset and average column is **bolded**, the second best is underlined. **Training time** reports the fine-tuning time of language models measured on the *Artery_Tibial* task.

Model	Time	eQTL Datasets and Average (AUROC)									
	(min)	Artery	Adipose	Fibro	Muscle	Nerve	Skin (NSE)	Skin (SE)	Thyroid	Blood	Average
Task-specific baselines											
Expert model[3]	-	0.741	0.736	0.639	0.621	0.683	0.710	0.700	0.612	0.689	0.681
CNN[3]	-	0.576	0.551	0.547	0.502	0.516	0.499	0.499	0.487	0.577	0.528
DNALongBench-official											
HyenaDNA[3]	210	0.479	0.513	0.584	0.487	0.511	0.471	0.544	0.529	0.512	0.514
Caduceus-Ph[3]	435	0.547	0.541	0.597	0.538	0.588	0.586	0.574	0.527	0.594	0.566
Caduceus-PS[3]	891	0.536	0.519	0.549	0.523	0.552	0.529	0.541	0.547	0.542	0.538
External-pretraining											
JanusDNA[6]	2520	**0.852**	0.769	**0.802**	0.864	**0.914**	**0.903**	0.846	**0.793**	0.821	**0.840**
NTv3-100M-pre[21]	117	0.805	0.744	0.698	0.798	0.885	0.886	0.727	0.778	0.808	0.792
Evo2-1B[20]	1304	0.745	0.725	0.691	0.802	0.816	0.831	0.705	0.698	0.771	0.754
Leakage-controlled-pretraining											
Mamba2	328	0.750	0.729	0.699	0.806	0.847	0.820	0.713	0.701	0.792	0.762
U-Net	**39**	0.756	0.733	0.680	0.795	0.851	0.880	0.699	0.735	0.792	0.769
H-Net-128BPT[10]	65	0.781	0.676	0.785	**0.880**	0.862	0.876	**0.862**	0.745	0.715	0.798
GeneZip-70M	50	0.845	**0.788**	0.720	0.877	0.829	0.886	0.820	0.781	**0.834**	0.820

Note. Due to resource limitations, Evo2-1B was fine-tuned using LoRA on 4×H100 GPUs, whereas all other models on a single A100 GPU.

B.2 eQTL prediction

We evaluate long-range variant effect modeling on the nine DNALongBench [3] eQTL classification tasks. Table 4 reports AUROC across tissues, along with end-to-end fine-tuning time under an identical pipeline; training time is measured on the Artery Tibial task.

Experimental settings. For GeneZip-70M, we fine-tune the cis-regulatory focused 12.8K checkpoint selected by average validation AUROC with the routing model frozen. This checkpoint uses the ratio targets promoter:CDS:UTR:exon:intron:NIG:DIG $= 1 : 16 : 2 : 4 : 2 : 2 : 4$. The allocation reflects a regulatory prior for this GTEx/Enformer-style benchmark: the dominant signal is expected to lie in noncoding cis-regulatory context around the variant and target promoter, so promoter, UTR/exonic boundary regions, introns, and nearby intergenic sequence retain higher resolution. This does not imply that CDS is generally uninformative for expression, since coding sequence can affect mRNA abundance through mechanisms such as nonsense-mediated decay, splicing, mRNA stability, or codon usage. To stabilize supervised adaptation on limited labeled data and isolate the learned compressor, we freeze the encoder, including the router and embedding layer, and optimize only the task head. We use AdamW with learning rate 2×10^{-4}, weight decay 0.1, $(\beta_1, \beta_2) = (0.9, 0.95)$, a linear learning-rate schedule with warmup ratio 0.1, and gradient clipping $\|g\|_2 \leq 1.0$. Because the input is long, we train for 3 epochs with per-device batch size 1 and gradient accumulation 8 (effective batch size 8 per device), using BF16 mixed precision.

In terms of efficiency, models with explicit downsampling/compression modules (NTv3 [21], H-Net, and GeneZip) fine-tune substantially faster than those without (HyenaDNA and Caduceus), underscoring the need to reduce effective sequence length before token mixing in long-context genomic tasks. GeneZip is substantially faster than HyenaDNA, Caduceus, and JanusDNA, completing training in 50 minutes compared to 210 minutes for HyenaDNA, 435/891 minutes for Caduceus variants, and 2520 minutes for JanusDNA, while remaining close to the fastest compressed baseline.

For effectiveness, GeneZip outperforms H-Net on 5 of 9 tissues, indicating that region-aware pretraining provides a transferable inductive bias beyond generic adaptive compression. GeneZip is competitive with the strongest baseline, JanusDNA: it achieves the best AUROC on Adipose and Blood, and the best or second-best AUROC on 6/9 tissues at displayed precision, with a 50.4× speedup (50 vs. 2520 minutes).

Table 5: Contact map prediction on DNALongBench [3]. We report stratum-adjusted correlation coefficient (SCC) and Pearson correlation (Corr) across five cell lines, along with the average. Best results are in **bold**; second best are underlined.

Model	HFF		H1hESC		GM12878		IMR90		HCT116		Avg	
	SCC ↑	Corr ↑	SCC ↑	Corr ↑	SCC ↑	Corr ↑	SCC ↑	Corr ↑	SCC ↑	Corr ↑	SCC ↑	Corr ↑
Task-specific baselines												
CNN-one-hot	0.0291	0.0571	0.0010	0.0297	0.0086	0.0321	0.0065	0.0302	0.0190	0.0203	0.0128	0.0339
External-pretraining												
JanusDNA	**0.1604**	**0.2125**	**0.1589**	**0.2099**	**0.1396**	**0.1872**	**0.1575**	**0.2066**	**0.1788**	**0.1993**	**0.1590**	**0.2031**
NTv3-100M-pre	0.1433	0.1951	0.1371	0.1873	0.1009	0.1548	0.1344	0.1832	0.1646	0.1825	0.1361	0.1806
Evo2-1B	0.0208	0.1005	0.0127	0.0918	0.0060	0.0922	0.0154	0.0912	0.0337	0.0742	0.0177	0.0900
Leakage-controlled-pretraining												
Mamba2	0.1122	0.1631	0.1157	0.1657	0.0803	0.1134	0.1101	0.1588	0.1358	0.1532	0.1108	0.1508
U-Net	0.1326	0.1853	0.1306	0.1823	0.1043	0.1543	0.1269	0.1782	0.1625	0.1812	0.1314	0.1762
H-Net-128BPT	0.1130	0.1626	0.1139	0.1597	0.0917	0.1405	0.1117	0.1597	0.1606	0.1811	0.1182	0.1607
GeneZip-70M	0.1324	0.1840	0.1338	0.1867	0.1071	0.1600	0.1310	0.1804	0.1618	0.1843	0.1332	0.1791

B.3 Contact map prediction

We then assess how different downsampling modules affect representation learning on a long-range genomics benchmark, the contact map prediction (CMP) task in DNALongBench [3]. CMP is a two-dimensional regression task that maps a 1,048,576-bp DNA window to a binned chromatin contact map at 2,048-bp resolution. Following prior work [28], we use a 2D CNN as the CMP predictor backbone, but vary its input features: mean-pooled bin embeddings produced by different sequence models. Specifically, we compare the following representations: one-hot DNA embeddings (CNN-one-hot); Evo2-1B [20] with precomputed coverage-aware pooled bin embeddings from the frozen backbone; NTv3-100M-pre; JanusDNA [6]; Mamba2 without compression; U-Net with uniform compression at 128 bp per token; H-Net-128BPT; and GeneZip-70M. For GeneZip-70M, we use the transcript-balanced 12.8K checkpoint selected by average validation SCC. This prior keeps promoter/CDS resolution high, UTR/exon resolution intermediate, and intronic/intergenic regions more compressed, which is plausible for chromatin contacts whose signal is distributed across genic and regulatory sequence.

For each cell-type subset (`HFF`, `H1hESC`, `GM12878`, `IMR90`, `HCT116`), we train the same lightweight CNN contact-map head on top of frozen sequence representations. The head uses projection dimension 128, distance-embedding dimension 16, a 2-layer 1D CNN with kernel size 5 and dropout 0.1, and a 3-layer 2D CNN with channels $[64, 64, 64]$, kernel size 3, dropout 0.1, and symmetrized outputs. We optimize with AdamW using learning rate 2×10^{-4}, weight decay 0, batch size 32, BF16 mixed precision, and gradient clipping $\|g\|_2 \leq 1.0$. We train for $5{,}000$ update steps.

Table 5 summarizes CMP results. JanusDNA is the strongest method on this task, reaching 0.1590/0.2031 average SCC/Corr. NTv3-100M-pre is the second strongest foundation-model baseline behind JanusDNA, reaching 0.1361/0.1806 average SCC/Corr. Frozen Evo2-1B embeddings transfer poorly in this setting, reaching only 0.0177/0.0900 average SCC/Corr; this is slightly above the one-hot baseline and far below long-context baselines with task-aligned tokenization or compression. GeneZip-70M reaches 0.1332/0.1791 average SCC/Corr, stays close to NTv3-100M-pre, and is the strongest compression model trained in this work on GM12878 (SCC/Corr) and HCT116 Corr. The region-ratio ablation in Table 10 indicates that RAR is most useful when its regional resolution prior is selected for the downstream validation objective.

B.4 Transcription initiation signal prediction

We further evaluate base-pair-resolution regulatory signal prediction on the DNALongBench transcription initiation signal prediction (TISP) task [3]. TISP predicts promoter-centered transcription initiation profiles for five experimental assays, making it a stringent test of whether a compressed sequence model can preserve fine-scale promoter information after long-context pretraining.

Evaluation protocol. We follow the official DNALongBench TISP task definition and evaluation protocol. We use 100,000 bp input sequences and 10 nucleotide-resolution output tracks, corresponding to plus and minus strands for FANTOM CAGE, ENCODE CAGE, ENCODE RAMPAGE,

Table 6: Transcription initiation signal prediction (TISP) on DNALongBench [3]. We report Pearson correlation (PCC) for FANTOM CAGE (FC), ENCODE CAGE (EC), ENCODE RAMPAGE (ER), GRO-cap (GC), and PRO-cap (PC), along with the average. Task-specific and DNALongBench-official baselines are copied from Cheng et al. [3]; all other language-model rows follow the official TISP evaluation protocol with chromosome-level post-hoc evaluation. Best overall scores are in **bold**; best non-expert scores are underlined.

Model	FC	EC	ER	GC	PC	Avg
Task-specific baselines						
Expert Model[3]	**0.808**	**0.710**	**0.749**	**0.624**	**0.774**	**0.733**
CNN[3]	0.029	0.038	0.043	0.037	0.066	0.042
DNALongBench-official						
HyenaDNA[3]	0.138	0.124	0.118	0.112	0.168	0.132
Caduceus-Ph[3]	0.114	0.088	0.088	0.097	0.154	0.109
Caduceus-PS[3]	0.113	0.088	0.090	0.102	0.156	0.108
External-pretraining						
JanusDNA[6]	0.003	-0.001	0.003	0.000	0.003	0.002
NTv3-100M-pre[21]	0.220	0.148	0.149	0.170	0.284	0.194
Evo2-1B (LoRA)[20]	-0.002	-0.002	-0.004	0.000	0.000	-0.002
Leakage-controlled-pretraining						
Mamba2[25]	0.336	0.248	0.240	0.264	0.382	0.294
U-Net[26]	-0.005	0.000	0.002	0.016	0.030	0.008
H-Net-128BPT[10]	0.241	0.193	0.186	0.220	0.335	0.235
GeneZip-70M	0.319	0.239	0.232	0.243	0.359	0.279

GRO-cap, and PRO-cap. All language-model rows evaluated in our pipeline use the same dense per-base regression head, yielding logits $\hat{z}_{i,t}$ interpreted as log-rates $\hat{\lambda}_{i,t} = \exp(\hat{z}_{i,t})$ for base position i and track t. Unless otherwise specified, we use full fine-tuning with batch size 1, gradient accumulation 4, AdamW learning rate 2×10^{-4}, weight decay 0.01, BF16 mixed precision, one epoch capped at 5,000 optimizer steps, validation every 1,000 steps, and seed 0. GeneZip disables the ratio loss during supervised TISP adaptation and is trained with the pseudo-Poisson KL objective

$$\mathcal{L}_{\text{TISP}} = \frac{1}{LT} \sum_{i,t} \left[y_{i,t} \log \frac{y_{i,t} + \epsilon}{\hat{\lambda}_{i,t} + \epsilon} + \hat{\lambda}_{i,t} - y_{i,t} \right], \tag{12}$$

with $\epsilon = 10^{-5}$.

Model selection uses the validation chromosome with the same chromosome-level post-hoc scoring path used at test time. Evaluation generates predictions for complete holdout chromosomes using 100 kb windows with a 50 kb step; only the center 50 kb of each 100 kb prediction is scored, and an end-aligned final window covers chromosome tails without double-counting overlapping segments. Positions near chromosome ends and unknown bases are excluded by the valid mask before computing Pearson correlations. For each assay, we combine plus and minus strand tracks and stream the sufficient statistics for PCC, then report the average over the five assays. Evo2-1B is evaluated with LoRA because model size and sharding constraints make full fine-tuning impractical.

For GeneZip-70M, we use the promoter-distal regulatory 12.8K checkpoint selected by average validation PCC. This prior preserves promoter resolution and also allocates high resolution to intronic and near-gene intergenic compartments. This is consistent with the TISP objective, where base-pair-resolution initiation profiles are dominated by promoter/TSS-proximal sequence features, while nearby cis-regulatory context may provide additional signal. Table 6 augments the DNALongBench TISP comparison with GeneZip-70M and additional baselines (H-Net-128BPT, Mamba2, U-Net, NTv3-100M-pre, JanusDNA, and Evo2-1B (LoRA)), using the same task head and chromosome-level post-hoc evaluation protocol with memory-compatible optimization budgets. The expert model remains a strong task-specific upper reference. Among non-expert sequence models, Mamba2 is strongest at 0.294 average PCC, while GeneZip-70M reaches 0.279 and improves substantially over

Table 7: AUROC and AUPRC for the enhancer-target gene prediction (ETGP) task. Higher is better. Best scores are in **bold**.

Model	AUROC	AUPRC
Task-specific baselines		
Expert Model[3]	**0.926**	0.407
CNN[3]	0.797	0.310
DNALongBench-official		
HyenaDNA[3]	0.828	0.249
Caduceus-Ph[3]	0.826	0.380
Caduceus-PS[3]	0.821	0.376
External-pretraining		
JanusDNA[6]	0.822	0.438
NTv3-100M-pre[21]	0.823	0.440
Evo2-1B[20]	0.737	0.095
Leakage-controlled-pretraining		
Mamba2[25]	0.783	0.322
U-Net	0.750	0.244
H-Net-128BPT	0.728	0.131
GeneZip-70M	0.823	**0.446**

the strongest published DNALongBench foundation-model baseline (HyenaDNA, 0.132) and over H-Net-128BPT under the same evaluation protocol (0.235). The validation-selected GeneZip prior preserves promoter-centered regulatory signal, and the uncompressed Mamba2 backbone remains a strong baseline for nucleotide-resolution TISP adaptation.

B.5 Enhancer-target gene prediction

We further assess whether GeneZip transfers to regulatory interaction prediction using the enhancer-target gene prediction (ETGP) task from DNALongBench [3]. For GeneZip-70M, we use the intergenic-focused 12.8K checkpoint selected by validation AUPRC. We interpret this prior as preserving intergenic/noncoding context for enhancer–target gene linking, not as evidence that distal-intergenic (DIG) sequence alone drives ETGP performance. DNALongBench restricts enhancer–gene candidates to within 450 kbp of the target TSS, whereas our NIG/DIG boundary is an operational annotation threshold. We fine-tune this checkpoint on the K562 dataset for 15 epochs with learning rate 1×10^{-3}, LoRA enabled, mean pooling, and best-checkpoint test loading. We also include H-Net and U-Net 12.8 Kbp controls evaluated with the same sequence-classification wrapper and mean pooling; based on held-out AUPRC, we report the frozen H-Net run and the unfrozen U-Net run. All other optimization and regularization settings follow the eQTL setup (AdamW, weight decay 0.1, $(\beta_1, \beta_2) = (0.9, 0.95)$, warmup ratio 0.1, gradient clipping at 1.0, BF16). We use per-device batch size 1 with gradient accumulation 8.

Table 7 reports AUROC and AUPRC. The Expert Model achieves the best AUROC (0.926), and GeneZip-70M attains the best AUPRC (0.446), outperforming JanusDNA by $+0.008$ (0.446 vs. 0.438) and the Expert Model by $+0.039$ (0.446 vs. 0.407). H-Net and U-Net remain below GeneZip-70M on AUPRC (0.131 and 0.244 vs. 0.446), indicating that uniform long-sequence readouts are less effective for sparse enhancer–gene ranking. AUROC differences among neural baselines are small (e.g., 0.823 for GeneZip-70M vs. 0.822 for JanusDNA), while the AUPRC gain shows that validation-selected intergenic/noncoding allocation improves the ranking of top-scoring enhancer–gene links. This supports the use of region-adaptive compression for cross-element regulatory association when the allocation prior matches the task.

C A Detailed Analysis of GeneZip's Compression Performance

Evaluation protocol. We evaluate the same encoder-only checkpoint set on human and mouse held-out data. For human, we tile the full held-out chromosomes `chr8`, `chr9`, and `chr10` into contiguous non-overlapping 128 Kbp windows. The raw block contains 3262 windows; using

the same standard ambiguity filter as the GENCODE preprocessing path (`max_n_frac=0.05`), we evaluate 3094 windows and skip 168 high-`N` or assembly-gap windows. For mouse, we use a fixed-seed three-chromosome block (`chr11`, `chr19`, and `chr10`) from the GENCODE-derived mouse genome and apply the same contiguous 128 Kbp tiling and ambiguity filter. The raw block contains 2456 windows; 2368 windows pass the filter and 88 high-`N` or assembly-gap windows are skipped. For each window, the encoder predicts a hard boundary mask (with reverse-complement averaging), inducing a variable-length tokenization; all metrics below are computed per window and then reported as mean $\pm$ standard deviation. For both H-Net and GeneZip, Table 8 reports the stage 2 boundary-mask BPT; stage 1 routing tracks are used only as diagnostics.

Metrics. Let a window contain L bases $x_{1:L}$ and let boundary positions $0 = s_1 < s_2 < \cdots < s_T < s_{T+1} = L$ define token spans $[s_t, s_{t+1})$. The *observed* base-per-token (BPT) is

$$\mathrm{BPT}_{\mathrm{obs}} \coloneqq \frac{1}{T}\sum_{t=1}^{T}(s_{t+1} - s_t) = \frac{L}{T}. \tag{13}$$

Given region labels, let B_r be the number of bases assigned to region r in the window, and let $N_r^\star$ denote the *target* BPT for region r (Eq. 5). The *expected* BPT under the target allocation is

$$\mathrm{BPT}_{\mathrm{exp}} \coloneqq \frac{\sum_r B_r}{\sum_r B_r / N_r^\star}, \tag{14}$$

and we report a global budget-matching ratio

$$\text{BPT-ratio} \coloneqq \frac{\mathrm{BPT}_{\mathrm{obs}}}{\mathrm{BPT}_{\mathrm{exp}}}. \tag{15}$$

Note that $\mathrm{BPT}_{\mathrm{exp}}$ is a base-count weighted harmonic mean of $\{N_r^\star\}$, and $\text{BPT-ratio} = (\sum_r B_r/N_r^\star)/T$ equals the ratio of expected-to-observed token counts.

To measure region-level budget adherence, we fractionally attribute each token to regions. Let $\mathcal{R}_r$ be the set of base indices labeled as region r, and define the (fractional) token count assigned to region r as

$$T_r \coloneqq \sum_{t=1}^{T} \frac{|[s_t, s_{t+1}) \cap \mathcal{R}_r|}{s_{t+1} - s_t}, \qquad \widehat{N}_r \coloneqq \frac{B_r}{T_r}. \tag{16}$$

We then define the micro-averaged alignment error

$$\mathrm{MicroErr} \coloneqq \frac{1}{\sum_r B_r}\sum_r B_r \left|\log \frac{\widehat{N}_r}{N_r^\star}\right|. \tag{17}$$

MicroErr is an absolute log error: $\exp(\mathrm{MicroErr})$ can be interpreted as the geometric mean multiplicative mismatch.

To characterize *where* token budget is spent, we report group-level enrichment over promoter, genic (CDS/UTR/exon), and intergenic (intron/NIG/DIG) regions:

$$\begin{aligned} \mathrm{Enrich}(g) &\coloneqq \frac{T_g/\sum_r T_r}{B_g/\sum_r B_r}, \\ T_g \coloneqq \sum_{r\in g} T_r, &\qquad B_g \coloneqq \sum_{r\in g} B_r. \end{aligned} \tag{18}$$

Here, $\mathrm{Enrich}(g) > 1$ indicates that the token budget is concentrated on group g relative to its genomic coverage. Finally, we report base-level perplexity

$$\mathrm{PPL} \coloneqq \exp\left(\frac{1}{L}\sum_{i=1}^{L} -\log p(x_i \mid x_{<i})\right). \tag{19}$$

Lower perplexity indicates better language modeling quality.

Table 8: Compressor analysis of encoder-only checkpoints on full-chromosome held-out blocks. Human uses contiguous windows from `chr8/chr9/chr10`; mouse uses contiguous windows from `chr11/chr19/chr10`. BPT-ratio is reported as a diagnostic of global token budget; PPL and MicroErr best values are **bold** within each species. Region enrichment values are interpreted relative to each row's region-allocation prior.

Species	Model	Observed BPT	BPT-ratio	PPL ↓	MicroErr ↓	Intergenic enrich	Promoter enrich	Genic enrich
Human	H-Net-BPT128	124.4 ± 4.4	0.972 ± 0.034	2.8699	-	1.001 ± 0.009	0.996 ± 0.138	0.970 ± 0.232
	GeneZip-12.8K-intergenic-focused	113.7 ± 16.1	1.397 ± 0.326	2.8618	0.4969	1.034 ± 0.038	0.668 ± 0.236	0.680 ± 0.247
	GeneZip-12.8K	189.2 ± 80.7	1.060 ± 0.240	2.8730	0.3372	0.922 ± 0.086	1.902 ± 1.041	1.593 ± 0.835
	GeneZip-128K	198.8 ± 96.0	1.106 ± 0.263	**2.8417**	**0.2676**	0.891 ± 0.127	2.424 ± 1.872	1.581 ± 1.001
Mouse	H-Net-BPT128	120.7 ± 5.2	0.943 ± 0.040	3.5559	-	1.005 ± 0.011	0.978 ± 0.178	0.912 ± 0.226
	GeneZip-12.8K-intergenic-focused	142.2 ± 18.4	1.538 ± 0.420	**3.5489**	0.4739	1.034 ± 0.041	0.621 ± 0.263	0.674 ± 0.253
	GeneZip-12.8K	162.7 ± 27.6	0.919 ± 0.199	3.5613	0.4189	0.923 ± 0.086	2.040 ± 1.115	1.682 ± 0.917
	GeneZip-128K	186.9 ± 33.3	1.054 ± 0.232	3.5586	**0.2566**	0.884 ± 0.133	2.832 ± 2.245	1.752 ± 1.270

Results and discussion. Table 8 evaluates the same learned compressors on full held-out human and mouse chromosome blocks. Across each species, the PPL values are close: the human rows range from 2.8417 to 2.8730, and the mouse rows range from 3.5489 to 3.5613. We therefore do not interpret this analysis primarily as a PPL ranking. Instead, it shows that the different routing policies preserve broadly comparable base-level reconstruction on thousands of contiguous held-out windows, even when their token budgets and regional allocations differ substantially.

Transferable region-aware compression. The main signal is the allocation pattern. H-Net-BPT128 serves as a uniform-budget baseline: its enrichment values stay close to one and it has no region-specific target, so MicroErr is not reported. GeneZip, by contrast, preserves the intended direction of the region-aware prior without using annotations at inference time. The intergenic-focused prior keeps intergenic enrichment above one and promoter/genic enrichment below one on both held-out human chromosomes ($1.034/0.668/0.680$ for intergenic/promoter/genic) and mouse chromosomes ($1.034/0.621/0.674$). Thus, the same learned policy transfers to unseen human chromosomes and to an unseen species while maintaining the specified intergenic-focused compression direction.

This transfer is not limited to one prior. The transcript-balanced GeneZip-128K model shows the opposite allocation direction in both species, compressing intergenic sequence more strongly while preserving denser promoter and genic resolution: its intergenic/promoter/genic enrichment is $0.891/2.424/1.581$ on held-out human chromosomes and $0.884/2.832/1.752$ on mouse chromosomes. Together, these results support the intended use of GeneZip as a controllable compression interface: different biological priors can specify where the model should spend token budget, and the learned routing policy retains that direction beyond the chromosomes and species seen during compressor training.

C.1 Repeat-subset predictability details

Table 9 reports the predictability measurements that support the redundancy-aware compression analysis in Section 4.2 and Figure 4. For each repeat subset, we report mean repeat-position PPL, matched-background PPL, and their ratio, averaged per window in the original full-window context. These values show that many TE-derived and tandem-repeat subsets are easier to predict than their matched background. This supports a compressibility prior for repeat interiors, not a claim that repetitive DNA is biologically irrelevant; TE-derived elements can be regulatory, and not every repeat subset is easier to predict than its matched background. The main-text BPT analysis then tests whether the router converts repeat predictability into stronger local compression while preserving resolution through overlapping gene-structure annotations when present.

D Case studies of GeneZip compression

To make the learned routing policy more concrete, we visualize the *boundary probability* (i.e., token-emission probability) along representative loci, together with a region taxonomy derived from GENCODE gene models [8]. Figure 5 overlays region labels (promoter/CDS/UTR/exon/intron/near-intergenic (NIG)) with per-base boundary probabilities produced by GeneZip-70M ($\mathrm{BPT} = 137.6$) and an H-Net-70M baseline ($\mathrm{BPT} = 122.2$) [10].

Table 9: Repeat-subset predictability details for the Figure 4 analysis. Each cell reports mean repeat-position PPL / matched-background PPL / PPL ratio, averaged per window in the original full-window context. PPL ratio below 1 means the named repeat subset is easier to predict than its matched background.

Species	Repeat subset	GeneZip repeat / bg / ratio	H-Net repeat / bg / ratio	GeneZip BPT ratio	H-Net BPT ratio
Human	DNA transposon	2.67 / 2.85 / 0.94	2.68 / 2.85 / 0.94	1.07	0.97
	LINE	2.17 / 2.95 / 0.74	2.17 / 2.95 / 0.74	1.17	0.92
	LTR	2.28 / 2.87 / 0.80	2.27 / 2.87 / 0.80	1.13	0.95
	SINE	1.69 / 2.94 / 0.58	1.69 / 2.94 / 0.58	2.67	1.24
	TR	1.97 / 2.87 / 0.69	2.02 / 2.86 / 0.71	1.56	0.79
Mouse	DNA transposon	2.97 / 3.58 / 0.83	2.98 / 3.57 / 0.83	2.00	1.18
	LINE	3.35 / 3.60 / 0.93	3.35 / 3.59 / 0.94	1.22	0.99
	LTR	3.61 / 3.58 / 1.01	3.61 / 3.56 / 1.02	1.08	0.99
	SINE	3.37 / 3.59 / 0.94	3.37 / 3.58 / 0.94	1.15	0.87
	TR	1.97 / 3.65 / 0.54	2.02 / 3.64 / 0.56	1.44	0.47

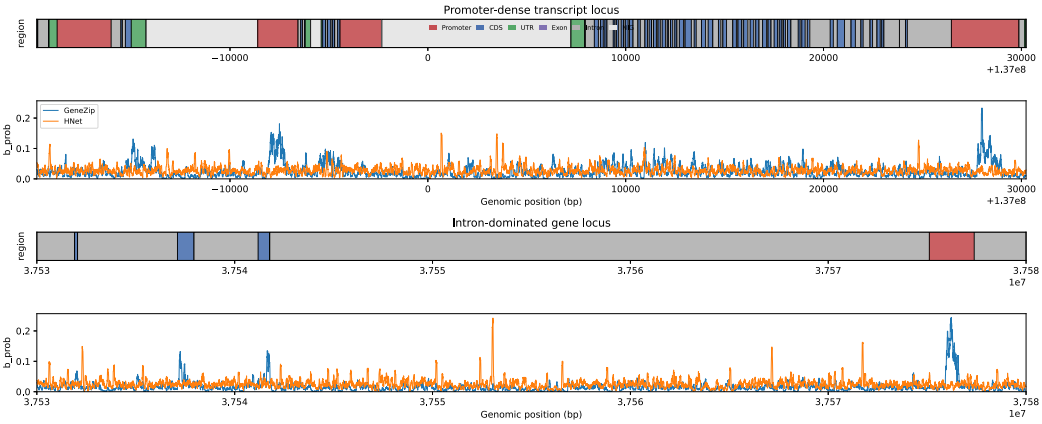


Figure 5: **Case studies of region-adaptive compression.** Region labels are induced from GENCODE gene models [8] (promoter/CDS/UTR/exon/intron/near-intergenic (NIG)), and the curves plot per-base boundary probability for H-Net [10] and GeneZip. We highlight two loci on chr9: a promoter-dense transcript locus (chr9:136,980,237–137,030,237; top) and an intron-dominated gene locus (chr9:37,530,000–37,580,000; bottom). Across both loci, GeneZip concentrates boundary mass in promoters and transcribed features, while suppressing diffuse boundaries in long intronic or intergenic spans.

Across both loci, GeneZip exhibits region-adaptive allocation: boundary probability increases in promoters and transcribed features (exon/CDS/UTR) and decreases over long intronic and intergenic spans, yielding higher effective resolution where genomic signals are dense. In the promoter-dense locus (top), GeneZip maintains elevated boundary density around successive promoter segments and nearby transcribed landmarks, while compressing relatively feature-sparse intervals between them. In the intron-dominated locus (bottom), GeneZip suppresses diffuse boundaries across the long gene body and intergenic spans, while preserving sharp peaks near promoters and exon-related landmarks. Compared to H-Net, which produces a noisier boundary profile with weaker alignment to region labels, GeneZip learns a more selective policy that better matches the Region-Aware Ratio objective.

E Ablation Studies

E.1 Region-Ratio Ablation Study

This section ablates the central control variable in GeneZip: the region multiplier μ_r in the Region-Aware Ratio (RAR) objective. Recall from Eq. 5 that the target base-pairs-per-token for region r is $N_r^\star = N\mu_r$. Thus, smaller μ_r assigns higher resolution to a genomic compartment, and larger μ_r

Table 10: Region-ratio ablation over the RAR multipliers μ_r. The region order is (promoter, CDS, UTR, exon, intron, NIG, DIG). Smaller μ_r gives higher resolution in that region, whereas larger μ_r gives stronger compression. eQTL reports average AUROC across tissues, ETGP reports AUROC/AUPRC, CMP reports average SCC/Corr across five cell lines, and TISP reports average PCC across five transcription-initiation assays.

Setting	μ_r	Biological allocation prior	eQTL AUROC	ETGP AUROC/AUPRC	CMP SCC/Corr	TISP PCC
Uniform RAR	$(1, 1, 1, 1, 1, 1, 1)$	no region-specific biological prior	0.8183	0.7931 / 0.2297	0.1182 / 0.1607	0.2349
Promoter-distal regulatory	$(1, 16, 8, 8, 2, 2, 4)$	promoter, intron, and near-gene intergenic resolution	0.7967	0.8204 / 0.4330	0.1324 / 0.1788	**0.2785**
Cis-regulatory focused	$(1, 16, 2, 4, 2, 2, 4)$	promoter, UTR, intron, and near-gene intergenic resolution	**0.8200**	0.8196 / 0.3554	0.1282 / 0.1751	0.2764
Intergenic-focused	$(32, 16, 8, 4, 2, 4, 1)$	intergenic/noncoding resolution; intentionally coarser promoter resolution	0.7863	**0.8225 / 0.4464**	0.1280 / 0.1735	0.2238
Transcript-balanced	$(1, 1, 2, 2, 8, 8, 16)$	promoter/CDS resolution with moderate UTR/exon resolution	0.7558	0.8160 / 0.3334	**0.1332 / 0.1791**	0.2741

compresses that compartment more aggressively. The ablation tests whether task-aware biological priors can steer GeneZip; it is not intended to identify a single universally optimal ratio.

The motivation is biological. Functional information is not uniformly distributed along the genome: GENCODE gene models provide stable gene-structure compartments, ENCODE and FANTOM annotations show that promoters, enhancers, and other cis-regulatory elements occupy specialized genomic contexts, and comparative constraint maps show that functional constraint is highly non-uniform across human DNA [7, 8, 23, 29, 30]. GeneZip uses this fact as a controllable compression prior. If a task depends on information concentrated in region family Y, we can lower μ_r for $r \in Y$ to spend more tokens there and increase μ_r elsewhere to keep the global budget feasible.

For final downstream reporting, the single GeneZip-70M row for each DNALongBench task is selected using validation performance among non-uniform 12.8K GeneZip RAR checkpoints. This means that the main summary reports validation-selected RAR priors per downstream task, not one fixed regional prior for all tasks. The transcript-balanced row in Table 10 provides the fixed-prior comparison. This 12.8K comparison keeps the pretraining-data sequence length and context-length budget matched to the leakage-controlled baselines. The uniform RAR setting is excluded from task-wise prior selection and retained as a region-agnostic RAR control with no region-specific multiplier. We select eQTL by average validation AUROC, ETGP by validation AUPRC, CMP by average validation SCC, and TISP by average validation PCC. This yields the cis-regulatory focused checkpoint for eQTL, intergenic-focused for ETGP, transcript-balanced for CMP, and promoter-distal regulatory for TISP; Table 10 reports the corresponding held-out test results.

Ablation settings. Table 10 reports five settings, always in the region order (promoter, CDS, UTR, exon, intron, NIG, DIG). The *transcript-balanced* prior is the default GeneZip prior: it preserves promoters and CDS at the highest resolution, keeps UTR/exon at intermediate resolution, and compresses intronic and intergenic sequence more strongly. The *cis-regulatory focused* prior preserves promoter, UTR, intronic, and near-intergenic sequence while compressing CDS more aggressively, reflecting that expression-modulating variants often act through noncoding cis-regulatory sequence [2, 31, 32]. The *promoter-distal regulatory* prior further emphasizes promoter, intron, and NIG sequence, while retaining moderate resolution in DIG, motivated by the mixture of promoter/TSS-proximal and nearby regulatory contexts captured by transcription-initiation and enhancer-target gene benchmarks [30, 33, 34]. The *uniform RAR* setting removes the region-specific multiplier but still uses the RAR framework, and is therefore a region-agnostic RAR control rather than a true no-RAR model. The *intergenic-focused* setting assigns high resolution to intergenic/noncoding sequence, with the highest resolution assigned to DIG and coarser resolution to promoters, testing whether performance follows the biological compartment emphasized by the task, instead of improving whenever the token budget is redistributed.

Transcription initiation signal prediction (TISP). The updated TISP results make the task-specific pattern clearer. TISP is promoter-centered: CAGE and RAMPAGE identify capped transcription start sites, while GRO-cap and PRO-cap capture nascent initiation signals. The relevant sequence signal is therefore expected to concentrate around promoters and TSS-proximal sequence, with nearby cis-regulatory context providing additional signal rather than replacing the promoter-centered objective [3, 29, 30, 35]. Consistent with this biology, the best TISP PCC is achieved by the promoter-distal regulatory prior (0.2785), which keeps promoters at maximal resolution ($\mu_{\text{promoter}} = 1$) and also allocates high resolution to intronic and near-gene intergenic regions. The cis-regulatory focused prior is close behind (0.2764), and the default transcript-balanced prior remains competitive (0.2741).

In contrast, uniform RAR reaches only 0.2349, and the intergenic-focused setting drops to 0.2238 because it compresses promoters with $\mu_{\text{promoter}} = 32$. This gap is biologically informative: TISP improves when GeneZip preserves compartments where initiation-associated regulatory grammar is expected to reside, not under arbitrary non-uniform compression.

Expression quantitative trait locus (eQTL) prediction. eQTL prediction is not a purely local promoter task. Regulatory effects can arise from promoters, UTRs, intronic enhancers, gene-body sequence, gene-adjacent noncoding sequence, and coding or splicing-linked mechanisms, while tissue-specific eQTL maps show that expression variation is distributed across diverse cis-regulatory contexts [7, 31, 32]. In this benchmark, the cis-regulatory focused prior obtains the best average AUROC (0.8200), with uniform RAR close behind (0.8183). The promoter-distal and intergenic-focused allocations drop to 0.7967 and 0.7863, and the transcript-balanced default reaches 0.7558 in the matched evaluation run. This trend differs from TISP: while transcription initiation benefits most from promoter and nearby regulatory resolution, this GTEx/Enformer-style eQTL benchmark benefits from a cis-regulatory allocation that preserves promoter, UTR, intronic, and near-gene intergenic sequence while compressing CDS more aggressively. The result does not imply that CDS is generally uninformative for expression. GeneZip's flexibility is useful because the regional prior can be matched to the biological structure of the downstream task; arbitrary non-uniform priors do not consistently improve performance.

Enhancer-target gene prediction (ETGP). ETGP is the ablation where noncoding intergenic and gene-proximal regulatory allocation matters most. Enhancer–gene links are mediated by enhancer activity, promoter activity, genomic distance, and physical or regulatory contact between candidate regulatory elements and target-gene-proximal sequence [33, 34, 36]. All non-uniform regulatory settings outperform the uniform RAR control in AUROC: promoter-distal regulatory reaches 0.8204, cis-regulatory focused reaches 0.8196, transcript-balanced reaches 0.8160, and the intergenic-focused prior reaches 0.8225, compared with 0.7931 for uniform RAR. The ETGP-selected setting supports the importance of preserving intergenic/noncoding sequence for enhancer–target gene linking. Because the benchmark restricts enhancer–gene candidates to within 450 kbp of the target TSS, and because our NIG/DIG boundary is an operational annotation threshold rather than a biological cutoff, we do not interpret this result as evidence that DIG alone drives ETGP performance. Instead, it indicates that task-specific allocation toward noncoding intergenic context is beneficial. The same prior is weak on eQTL and TISP, so its ETGP performance should be read as task-specific alignment, not as a globally superior compression policy.

Chromatin contact map prediction (CMP). CMP is a stress test for 1-Mb local chromatin folding, not a single-region local signal. At this scale, contact maps include CTCF-associated loops, TAD/sub-TAD organization, enhancer–promoter contacts, Polycomb-like contacts, and local compartment-like boundary signals [28, 36–39]. Here, every non-uniform GeneZip setting improves over the uniform RAR control (0.1182 SCC): promoter-distal regulatory reaches 0.1324, cis-regulatory focused reaches 0.1282, intergenic-focused reaches 0.1280, and transcript-balanced reaches 0.1332. The small spread among biologically plausible priors is expected for a structure-prediction task whose signal is distributed across compartments. The important result is that task-aware region allocation remains competitive under a two-dimensional long-range task and also helps beyond local promoter or variant tasks.

Takeaway. The RAR ablation suggests that no single regional compression schedule is optimal across long-range genomics tasks. TISP favors high resolution around promoter/TSS-proximal and nearby regulatory sequence, consistent with base-pair-resolution transcription initiation profiling. eQTL favors cis-regulatory allocation, consistent with sequence-based classification of expression-modulating variants near target genes. ETGP benefits from preserving intergenic/noncoding context, but because the benchmark restricts candidates to within 450 kbp of the target TSS and our NIG/DIG boundary is operational, we interpret this as evidence for noncoding intergenic allocation rather than DIG-specific biology. CMP favors a more balanced transcript-aware allocation, consistent with the mixed sequence determinants of 1-Mb local chromatin folding, including CTCF-associated loops, TAD/sub-TAD organization, and cell-type-specific regulatory contacts. These results support task-aligned regional compression priors. However, because different RAR priors can also change the observed number of retained tokens, we treat the ablation as evidence for biologically plausible

Table 11: Auxiliary context-length ablation on DNALongBench under the default transcript-balanced GeneZip prior. Both rows use 70M GeneZip checkpoints and matched downstream fine-tuning/evaluation protocols. eQTL reports average AUROC across nine tissues, ETGP reports AUROC/AUPRC, CMP reports average SCC/Corr across five cell lines, and TISP reports average PCC across five transcription-initiation assays.

Model	Context	eQTL AUROC	ETGP AUROC/AUPRC	CMP SCC/Corr	TISP PCC
GeneZip-70M	12.8K	0.7558	0.8160 / 0.3334	0.1332 / 0.1791	0.2741
GeneZip-70M-128K	128K	0.7837	0.8354 / 0.4559	0.1360 / 0.1823	0.2993

Table 12: Statistics of the GENCODE-derived pretraining corpora used in this work. **Total BP** is reported in Mbp. Region columns report Mbp with within-split percentages in parentheses under our 7-region taxonomy: promoter, coding sequence (CDS), UTR, exon, intron, near-intergenic (NIG), and distal-intergenic (DIG). **N BP** reports ambiguous bases (`N`) in Mbp. The `test` split corresponds to contiguous, non-sampled windows from held-out chromosomes (`chr8`, `chr9`, `chr10`), while `train`/`valid` are sampled windows.

Dataset	Split	Total BP (M)	N BP (M)	Promoter	CDS	UTR	Exon	Intron	NIG	DIG
GENCODE-Human-12.8k	train	12160.00	0.16	1815.54 (14.93%)	324.84 (2.67%)	296.16 (2.44%)	190.66 (1.57%)	6758.93 (55.58%)	2636.61 (21.68%)	137.26 (1.13%)
	valid	640.00	0.01	96.56 (15.09%)	17.08 (2.67%)	15.51 (2.42%)	9.89 (1.55%)	355.94 (55.61%)	138.24 (21.60%)	6.79 (1.06%)
GENCODE-Human-128k	train	12672.00	0.63	1047.20 (8.26%)	223.82 (1.77%)	259.47 (2.05%)	174.46 (1.38%)	7546.70 (59.55%)	3266.07 (25.77%)	154.28 (1.22%)
	valid	128.00	0.01	10.52 (8.22%)	2.23 (1.74%)	2.72 (2.12%)	1.79 (1.40%)	77.64 (60.65%)	32.21 (25.17%)	0.90 (0.70%)
Held-out (chr8, 9, chr10)		417.34	17.52	22.44 (5.38%)	3.42 (0.82%)	4.81 (1.15%)	5.11 (1.22%)	243.77 (58.41%)	117.50 (28.15%)	20.31 (4.87%)

allocation effects rather than a fully token-count-matched causal test; fully isolating regional allocation from token-count effects requires matched-BPT controls.

E.2 Context-Length Ablation Study

The region-ratio study above changes the region-allocation prior at fixed 12.8 Kbp context to keep the main comparison strictly fair with the leakage-controlled baselines, which use the same context length and pretraining-data sequence length. Due to computation budget, we did not train 128K versions for every baseline model or every region-ratio prior. We therefore treat Table 11 as an auxiliary context-length ablation under the default transcript-balanced prior, not as a replacement for the main task-selected 12.8K comparison.

The comparison shows a consistent benefit from length scaling. At the same 70M-parameter model size, continuing pretraining from 12.8K to 128K contexts improves every downstream task under this auxiliary setup: eQTL average AUROC increases from 0.7558 to 0.7837, ETGP improves from $0.8160/0.3334$ to $0.8354/0.4559$ AUROC/AUPRC, CMP improves from $0.1332/0.1791$ to $0.1360/0.1823$ SCC/Corr, and TISP PCC increases from 0.2741 to 0.2993. The longer-context pretraining stage improves language-modeling metrics and can transfer into better downstream performance when compute allows matched 128K variants.

F Training Data Details

We construct the human pretraining corpora from the GRCh38 primary assembly together with the GENCODE v49 *basic* gene annotation [8].

Region taxonomy and priority overlay. We build a genome-wide region map and label each base with exactly one region under a deterministic priority order: `promoter` $>$ `cds` $>$ `utr` $>$ `exon` $>$ `intron` $>$ `nig` $>$ `dig`. Promoters are defined as symmetric windows around the transcript TSS (TSS ± 1 kbp). We further partition intergenic sequence into near-intergenic (NIG) and distal-intergenic (DIG) based on proximity to annotated transcription start sites (TSSs). After applying the region-priority overlay (so that promoter/genic regions take precedence), an intergenic base is labeled as NIG if it lies within 450 kbp (genomic distance) of any annotated TSS; otherwise it is labeled as DIG.

Sampling, splits, and formats. We create two corpora with fixed context lengths: 12.8K bp and 128K bp. For `train/valid`, windows are sampled with a mixture of center choices that increases coverage near transcription start sites and coding regions, alongside uniform sampling across the remaining genome. Each window stores the raw DNA sequence together with a run-length encoded segmentation over the 7-region taxonomy. The `valid` split is an in-distribution holdout constructed with the same sampling procedure as `train`.

DNALongBench leakage filtering and held-out chromosomes. To tailor the corpus for DNALongBench evaluation and mitigate benchmark leakage [3, 40], we remove all genomic intervals used by DNALongBench validation/test sets from the sampling pool for `train/valid`. In addition, we fully hold out the DNALongBench transcription-initiation held-out chromosomes `chr8`, `chr9`, and `chr10` during sampling. These chromosomes are reserved to construct a chromosome-level `test` split by tiling `chr8`, `chr9`, and `chr10` into contiguous, non-overlapping windows of the target context length (without sampling).

G Comparison to Tokenization Methods

Tokenization can be interpreted as a special case of the encoder in Eq. 1: it maps a raw sequence $x_{1:L}$ into a sequence of units $u_{1:T}$ before token mixing. GeneZip studies a different problem from tokenization. Most DNA tokenization methods are designed to define useful sequence units or vocabularies. GeneZip targets high-ratio, controllable length reduction for long-context inputs. This distinction matters because our target setting requires compressing 10^5–10^6 bp sequences into a much shorter latent sequence while preserving high resolution in biologically important regions.

BPE tokenization. Byte-Pair Encoding (BPE) learns a static merge table from corpus-level substring frequencies and then applies the same merge rules to every sequence [41]. DNABERT-2 introduces BPE tokenization for DNA foundation models, where it provides a compact vocabulary-level representation for multi-species genomes [17]. However, BPE does not optimize for an explicit long-context compression budget. Its merge decisions are frequency driven, global, and fixed after tokenizer training; they do not enforce a per-sample BPT target, a bounded routing budget, or a region-wise allocation constraint such as $\widehat{N}_r \approx N_r^\star$. BPE changes the input representation, yet it does not directly serve GeneZip's goal of high-ratio, region-aware compression for long-context sequence modeling.

Learnable Tokenization. Recent DNA tokenization work also learns sequence units without a hand-designed vocabulary. MxDNA learns tokenization through a sparse mixture of convolution experts and deformable convolution, targeting discontinuous, overlapping, and ambiguous genomic segments [11]. MergeDNA frames dynamic tokenization as token merging: it stacks differentiable token-merging blocks with local-window constraints to chunk adjacent bases, then couples the dynamic tokenizer with latent Transformer modules and context-aware pretraining objectives [12]. Table 13 includes the available tokenization baselines for comparison. BPE is a drop-in tokenizer, so we evaluate it with the same backbone and GENCODE training data as GeneZip; it reaches only 1.4 bp/token. Official MxDNA is not drop-in because its learned tokenization is a model-internal conversion layer in a Transformer, changing hidden-state length and attention masks. A faithful Mamba implementation would require reimplementation and retraining, so we use the authors' released checkpoint; it gives around 3.0 bp/token. At this resolution, training on 450K-bp eQTL/ETGP windows remains infeasible for MxDNA under the same budget, leading to CUDA out-of-memory errors. GeneZip reaches about 170 bp/token on these windows, enabling the downstream speed and scale advantage. The point of this comparison is scope: BPE and MxDNA are useful tokenizers, while GeneZip is designed for the case where long-context compute is the bottleneck and high-ratio region-aware compression is required.

H Limitations and Future Work

The current GeneZip pre-training is derived on human genomic data. The full-chromosome human/mouse compressor analysis in Table 8 provides inference-time evidence that the learned routing policy transfers across held-out chromosomes and to another mammalian genome while still concentrating token budget according to the selected biological prior. This does not substitute for full

Table 13: Comparison with tokenization baselines on long-context downstream windows. BPT is evaluated on the eQTL Artery Tibial test set and computed as total bp divided by total tokens, using effective stage 2 router tokens for GeneZip. Downstream metrics follow Table 2.

Model	BPT	eQTL AUROC	ETGP AUPRC
BPE	1.3947	0.7544	0.2540
MxDNA	3.0283	OOM	OOM
GeneZip	169.6499	0.8200	0.4464

multi-species pretraining and downstream benchmarking. Extending the training setup to broader multi-species resources with harmonized region annotations is an important direction for future work.

GeneZip builds on the underlying H-Net routing architecture [10]. Our contribution is to introduce a region-aware compression objective and bounded budget control on top of this dynamic routing framework. The boundary predictor still relies on representation dissimilarity between neighboring tokens, and its decisions may therefore be affected by noisy or low-complexity DNA sequences where local dissimilarity does not always correspond to functionally useful resolution.

Finally, the current supervision is centered on static region annotations and the Region-Aware Ratio objective. A promising next step is to incorporate richer function-oriented supervision beyond annotation labels, for example sequence-to-function objectives inspired by models such as Enformer and NT-family genomic predictors [1, 21]. We do not anticipate direct negative societal impact from the present methodological contribution. These technical scope limitations should still be considered when applying GeneZip-style compression to new organisms, datasets, or downstream genomic prediction settings.