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]University of Ottawa
[4]National Research Council of Canada [5]HEC Montréal
Project page: `andyjzhao.github.io/GeneZip/`

Abstract

Long-context DNA modeling is computationally expensive, motivating existing approaches to reduce sequence length through uniform downsampling. However, functional information and sequence redundancy are unevenly distributed across the genome, calling for finer resolution in functionally dense regions and stronger compression of redundant sequence. We introduce GeneZip, a region-aware DNA compression framework that learns where to allocate a limited token budget. Building on H-Net-style dynamic routing, GeneZip combines a Region-Aware Ratio objective, which uses regional labels derived from GENCODE gene annotations to specify compression targets during training, with bounded routing for stable token counts. At inference, it compresses raw DNA without annotations. GeneZip provides three main benefits. First, it preserves predictive utility under high compression: GeneZip-70M achieves 137.6-fold compression with a validation perplexity increase of 0.31 relative to uncompressed Mamba2-84M, and the best average rank across four DNALongBench tasks among compression models with leakage-controlled pretraining. Second, it learns region-aware allocation with emergent repeat-aware compression. Alternative regional priors yield task-dependent performance trade-offs. Without repeat-specific supervision, GeneZip compresses transposable-element-derived and tandem repeats more strongly than matched non-repeat sequence; its tandem-repeat compression preference transfers to a nine-species evaluation panel without further training. Third, it enables efficient long-context modeling: on 450-kb inputs, GeneZip-70M achieves 36.1-fold higher inference throughput than JanusDNA-7.7M on a single H100 GPU.

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 downsampling [1, 2]. These strategies make long inputs feasible while allocating resolution according to architecture instead of genome structure. Evolution shapes uneven functional information density across DNA. On the one hand, evolutionarily constrained bases are enriched in coding exons, UTRs, and promoters relative to intergenic sequence [7]. On the other hand, transposable-element expansion and tandem repetition introduce redundancy [8]. This motivates finer resolution for functionally dense regions and stronger compression of redundant sequence.

[*]Equal contribution. Correspondence to Jian Tang.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

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 [9], while adding biological control over where the token budget is spent. Specifically, its Region-Aware Ratio (RAR) objective trains the router toward region-wise sequence compression 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 shortens the sequence processed by the backbone by 137.6-fold, with a +0.31 perplexity increase relative to uncompressed Mamba2. It achieves the best average rank across four DNALongBench downstream tasks among compression models with leakage-controlled pretraining. These tasks cover contact map prediction, eQTL prediction, enhancer–target gene prediction, and transcription-initiation signal prediction.

• **Region-aware compression:** Our experiments show that GeneZip achieves high, region-aware compression with emergent repeat-aware allocation. GeneZip compresses unseen sequences without annotations at inference. Post-hoc RepeatMasker/TRF analyses reveal stronger local compression of TE-derived interspersed repeats and tandem-repeat interiors than matched non-repeat sequence, without repeat supervision.

• **Efficient long-context modeling:** On 450-kb inputs, 70M GeneZip achieves a 36.1× inference speedup over the uncompressed JanusDNA-7.7M model [6] (Table 2; Appendix B.2). By reducing the effective token-mixing length, GeneZip also supports 128K-context and 636M-parameter pretraining on a single H100 80GB GPU.

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 [9]. 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 [10–13]. GeneZip shares this view of learning the units presented to the backbone, with a focus on extreme, controllable sequence-length compression for long-context DNA. Concurrent dnaHNet applies H-Net's hierarchical dynamic chunking to DNA, retaining its ratio loss with stage-wise global compression targets [14]. GeneZip introduces region-specific budget supervision to control where resolution is retained, together with bounded routing to constrain token counts.

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, 15–19]. A complementary line of sequence-to-function models, including Enformer, AlphaGenome, and NTv3, uses uniform downsampling to predict regulatory tracks and variant effects over hundreds of kilobases to megabase context [1, 2, 20]. 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. Long-context genomic models such as NTv3, Enformer, and AlphaGenome reduce sequence length through architecture-defined downsampling schedules [1, 2, 20]. These schedules apply the same reduction across genomic regions, although the resolution needed to preserve useful information varies with region and task: coding exons occupy only $\sim$1–2% of the human genome and are enriched for evolutionary constraint [7, 8], while regulatory signal is distributed across promoters, UTRs, introns, enhancers, and distal intergenic regions in a task-dependent manner. GeneZip instead uses region-aware ratio supervision to allocate finer resolution to selected genomic regions while retaining content-dependent boundaries. Bounded routing constrains the total token count, making this allocation controllable even at high compression. Our experiments show that GeneZip achieves high, region-aware compression with emergent repeat-

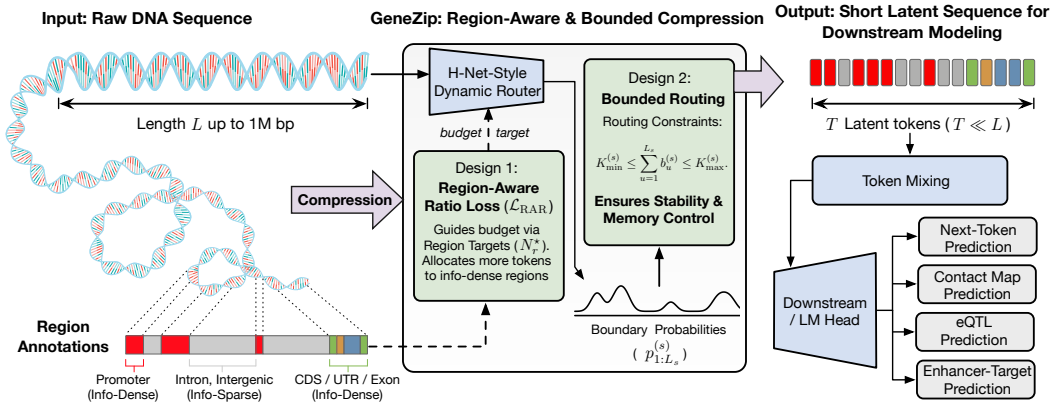


Figure 1: GeneZip overview. GeneZip compresses long-context DNA into a much shorter latent token sequence via hierarchical dynamic routing. During training, the Region-Aware Ratio (RAR) loss learns a region-adaptive token budget, and bounded routing enforces global token-count constraints for stable 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.

aware allocation and the best average rank across four DNALongBench downstream tasks among compression models with leakage-controlled pretraining.

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 [21] or long-sequence 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 [9]; the Region-Aware Ratio (RAR) loss supervises region-wise token budgets from gene-structure annotations (CDS, UTR, exon, intron, promoter, and intergenic regions). 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 [9]. At each stage, the encoder (i) adaptively partitions a long sequence into variable-length chunks and (ii) selects a boundary representation for each chunk, 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 [9], 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:

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

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

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

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. The boundary representation of each chunk is retained, producing a shorter sequence of length $L_{s+1} = \sum_{t=1}^{L_s} b^{(s)}_t$. The selected sequence, denoted as $h^{(s+1)}_{1:L_{s+1}}$, serves as input to the next stage. 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 [9]. 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 labels are never used as supervision; they are used only post hoc to test whether the learned router compresses repetitive DNA more strongly than matched non-repeat sequence.

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

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

Intuitively, smaller μ_r means *higher* resolution (more tokens) and larger μ_r means *stronger* compression. Promoters and genic elements are enriched for evolutionary constraint [7], motivating finer resolution in these regions. By default, we set the multipliers to: promoter $= 1$, CDS $= 1$, UTR $= 2$, exon $= 2$, intron $= 8$, near-gene intergenic $= 8$, distal-intergenic $= 16$. We compare alternative pretrained allocation priors across four downstream tasks in Appendix E.1.

Region-Aware Ratio (RAR) loss. RAR guides each region toward its target compression across S routing stages. Because successive stages compound the reduction in sequence length, we split the region target $N_r^\star$ (Eq. 5) into equal multiplicative factors. This gives each stage a target compression and corresponding fraction of positions to retain:

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

Applying these targets requires a region label for each position at every stage. The first stage uses the base-level annotations $r^{(1)}(t) \in \mathcal{R}$; subsequent stages inherit the labels of the boundary-selected representatives from the preceding stage. Specifically, if $i_t^{(s-1)}$ indexes the t-th position retained by $b^{(s-1)}$, then $r^{(s)}(t) = r^{(s-1)}(i_t^{(s-1)})$. These propagated labels are used only for RAR supervision. For the $L_r^{(s)} \triangleq \sum_{t=1}^{L_s} \mathbf{1}\{r^{(s)}(t) = r\}$ positions assigned to region r, we track both the fraction actually retained, $F_r^{(s)}$, and the mean boundary probability, $G_r^{(s)}$. The first measures the realized allocation; the second provides a differentiable quantity through which the loss can adjust the router:

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

To adjust the boundary probabilities according to the realized allocation's deviation from the target, we couple these two statistics using the ratio loss:

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

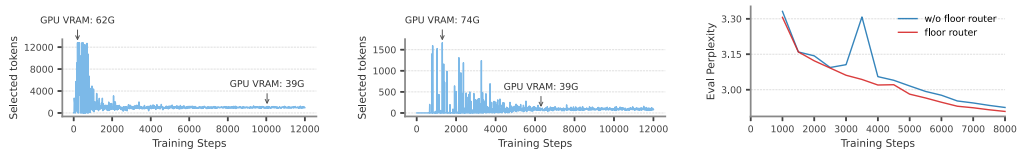
Intuition. RAR couples the realized keep rate $F_r^{(s)}$ after projection with its expectation $G_r^{(s)}$ before projection to match $\tau_r^{\star(s)}$. When $F_r^{(s)} = G_r^{(s)}$, Eq. (8) is minimized at $F_r^{(s)} = G_r^{(s)} = \tau_r^{\star(s)}$ with $\mathcal{L}_{\mathrm{RAR}}^{(r,s)} = 1$. If too many positions are retained, the loss pushes the regional boundary probabilities down; if too few are retained, it pushes them up. We weight each regional loss by its share of positions at that stage and sum across stages, so each region contributes in proportion to its representation:

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

3.3 Bounded Routing for Stable Training

GeneZip learns how to compress DNA through the RAR loss, progressively approaching the target regional compression ratios. During this learning process, however, the number of retained tokens can fluctuate substantially: insufficient compression can produce too many tokens and cause GPU memory spikes, whereas excessive compression can leave too few tokens for stable learning (Figure 2). We therefore introduce bounded routing, which constrains the token count at each stage to a prescribed interval, using a ceiling to limit memory usage and a floor to prevent excessive token reduction.

Reference token budget. RAR specifies how token density should vary by region; bounded routing constrains the total token count permitted at each stage. We use the base BPT anchor N to define a reference keep rate $\tau^{(s)} = N^{-1/S}$, with $\prod_{s=1}^{S} \tau^{(s)} = 1/N$. This reference sets the scale for the allowed interval, while the region-specific targets $N\mu_r$ continue to guide the learned allocation.



(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 perplexity trajectories compared to training without it.

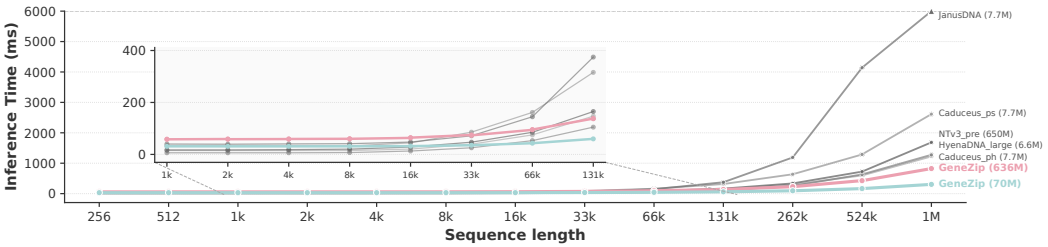


Figure 3: **Single-sequence inference latency on long-context inputs.** Mean forward-pass latency versus input length (256 bp to 1M bp); measurement details are in Appendix B.1. GeneZip (70M/636M) maintains low latency across the measured range. The inset shows the 1K–131K range.

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 [9], we train GeneZip end-to-end with an autoregressive 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 shows that GeneZip maintains low inference latency as input length increases to 1M bp, with a widening latency advantage over the baseline models at longer contexts. This scaling behavior supports its use for long-context DNA modeling. Benchmark details are provided in Appendix B.1, and Section 4.3 reports complementary results on real genomic sequences.

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 high-compression methods (U-Net, H-Net-70M, and GeneZip). Underline indicates the best PPL among 70M high-compression models trained at 12.8K context 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-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

We pretrain GeneZip on 100B base pairs sampled from the human GRCh38 reference genome with a 12.8K-bp context. Regional labels are derived from GENCODE gene annotations [22] to supervise region-aware compression. We set the regional multipliers to $\mu_r = (1, 1, 2, 2, 8, 8, 16)$ for promoter, CDS, UTR, exon, intron, near-intergenic (NIG), and distal-intergenic (DIG) regions, respectively. Unless stated otherwise, GeneZip denotes the 70M model; the 636M and 128K suffixes identify larger-model and longer-context variants, respectively. Pretraining and data details are provided in Appendices A.1 and F.

We evaluate whether high compression preserves useful information, how the router allocates tokens, and whether the resulting representations support long-range prediction. Section 4.1 measures reconstruction quality and its response to model scaling and continued long-context pretraining; Section 4.2 examines emergent compression of repetitive DNA; and Section 4.3 evaluates downstream predictive utility.

The appendix develops these findings along three complementary directions. To characterize predictive performance alongside computational cost, Appendices B and C report inference scaling and speed, together with complete task-level results. Appendix D examines GeneZip's compression behavior, including regional budget consistency and preference transfer, repeat-aware compression across species, and routing at individual loci. Appendix E then examines how alternative regional priors change downstream trade-offs. Finally, Appendix G compares GeneZip with BPE and MxDNA to assess compression and long-context prediction relative to existing tokenization methods.

4.1 Pretraining analysis: compression quality and scaling

GeneZip and the pretraining baselines share a Mamba2 token-mixing backbone [23], with comparable parameter counts (70M–84M) in the main comparison. Mamba2 without compression serves as the uncompressed baseline; U-Net [24] and H-Net [9] provide uniform and learned compression baselines, respectively. Our H-Net-70M baseline follows the same dynamic-chunking and global ratio-loss formulation used by dnaHNet [14], instantiated with a Mamba2 backbone and a target of 128 bp/token. We train it alongside GeneZip on the same corpus, context length, and pretraining budget to compare global and region-aware compression under a common training setup. The remaining pretraining baselines also share these training conditions. Larger and longer-context GeneZip variants test scaling (Appendix A.1). We report overall and region-wise validation perplexity (PPL).

Table 1 establishes the reconstruction cost of high compression. Uncompressed Mamba2 and the lightly compressed H-Net-3BPT both achieve an overall PPL of approximately 2.41. At compression levels above 100 BPT, H-Net and U-Net have higher PPLs of 2.7930 and 2.8414, respectively.

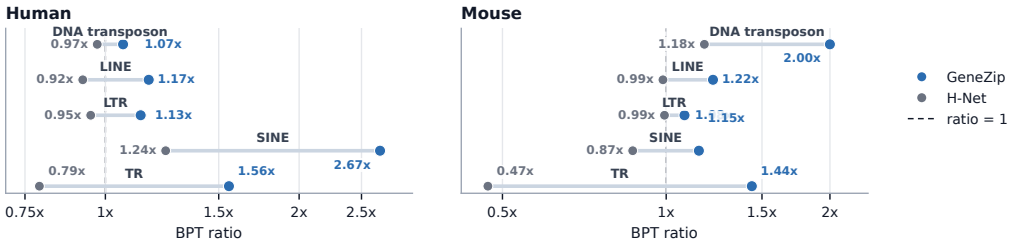


Figure 4: **Repeat-aware compression across TE-derived and tandem-repeat sequence.** Repeat positions are compared with matched non-repeat positions in the same full-window context. BPT ratio above 1 means stronger local compression. GeneZip has higher repeat/non-repeat BPT ratios than H-Net in every displayed subset, despite nearly identical repeat-subset PPL (Table 10). H-Net shows weaker and less consistent repeat-specific compression.

GeneZip reduces this reconstruction penalty while using fewer tokens. Among the comparable-size high-compression models trained at 12.8K context, it achieves the lowest overall and all reported region-wise PPLs at the highest BPT (137.6). Its overall PPL of 2.7259 is only 0.31 above uncompressed Mamba2, improving the compression–reconstruction trade-off over H-Net and U-Net.

Overall PPL also improves with model size and continued long-context pretraining. At 12.8K context, the 636M model lowers PPL from 2.7259 to 2.6620 at a similar BPT. After a further 100B-base pretraining stage at 128K context (Appendix A.1), the 70M model reaches 2.6557 PPL at 169.2 BPT, narrowing the gap to Mamba2 to 0.24. GeneZip-636M-128K achieves the lowest overall PPL among the high-compression models (2.6484).

4.2 Repeat-aware Compression

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 [25, 26]. 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 held-out human and mouse sequences (Appendix D.2). 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.

Repeat regions offer greater scope for compression because they are generally easier to predict than the surrounding background. Both GeneZip and H-Net have lower PPL on repeats than on matched non-repeat positions in nine of the ten human/mouse subsets (Table 10). For tandem repeats, the repeat/non-repeat PPL ratios are 0.69–0.71 in human and 0.54–0.56 in mouse, showing that both models find these regions substantially more predictable than the background.

GeneZip exhibits an emergent **repeat-aware compression** behavior: it assigns fewer tokens per base to repetitive regions than to the background without repeat-specific supervision. Its training labels describe gene structure rather than repeat classes. Despite nearly identical repeat-subset PPL, GeneZip has higher repeat/non-repeat BPT ratios than H-Net in all ten comparisons (Figure 4). For tandem repeats, GeneZip's ratios are 1.56 in human and 1.44 in mouse, compared with 0.79 and 0.47 for H-Net: GeneZip compresses these repeats more strongly than the background, whereas H-Net compresses them less. This emergent allocation behavior also transfers beyond the human training genome, with higher TR/non-TR BPT ratios than H-Net in all nine species of a separate evaluation panel (Appendix D.2).

4.3 DNALongBench downstream evaluation

We evaluate whether high compression preserves downstream predictive utility by adapting the same pretrained GeneZip checkpoint separately to four DNALongBench [3] tasks: contact map prediction (CMP), eQTL prediction, enhancer-target gene prediction (ETGP), and transcription initiation signal prediction (TISP). For GeneZip, Mamba2, H-Net, and U-Net, we exclude DNALongBench

Table 2: DNALongBench results and inference cost. BPT reports validation compression; measurement protocols are detailed in Appendix C. *JanusDNA-7.7M uses the architectural value implied by its published single-nucleotide representation. Inference reports total time for 100 sequences of 450 kb; hardware, batching, and timing details are in Appendix B.2. †Evo2 runs out of memory on one GPU. Average rank averages the four descending task ranks among all 10 rows, using displayed scores and average ranks for ties; inference time is excluded. Best task scores and lowest average rank within *Leakage-controlled-pretraining* are **bold**; best across all rows are underlined.

Model	BPT	Inference (s) ↓	CMP SCC	eQTL AUROC	ETGP AUPRC	TISP PCC	Avg. rank ↓
DNALongBench-official							
HyenaDNA-6.6M[4]	1.0	22.86	0.115	0.514	0.249	0.132	8.00
Caduceus-Ph-7.7M[5]	1.0	35.06	0.133	0.566	0.380	0.109	5.63
Caduceus-PS-7.7M[5]	1.0	68.36	0.127	0.538	0.376	0.108	7.00
External-pretraining							
JanusDNA-7.7M[6]	1.0*	228.63	0.159	0.840	0.438	0.002	3.25
NTv3-100M-pre[20]	127.99	13.27	0.136	0.792	0.440	0.194	2.75
Evo2-1B[19]	1.0	—†	0.018	0.754	0.095	-0.002	9.25
Leakage-controlled-pretraining							
Mamba2-84M[23]	1.0	40.80	0.111	0.762	0.322	0.294	6.25
U-Net-70M[24]	127.99	4.85	0.131	0.807	**0.390**	0.178	4.00
H-Net-70M[9]	122.19	11.78	0.118	0.791	0.225	0.303	5.75
GeneZip-70M	137.59	6.34	**0.133**	**0.817**	0.375	**0.336**	**3.13**

validation/test intervals and hold out chromosomes 8, 9, and 10 from the shared pretraining corpus. Table 2 separates this leakage-controlled group from externally pretrained models and official benchmark results. For eQTL, ETGP, and TISP, GeneZip and H-Net use a common routing ceiling of 70,000 tokens per stage with no additional floor. Task- and model-specific adaptation settings and measurement protocols are provided in Appendix C.

Performance. At 137.6 BPT, GeneZip achieves the best average rank among compression models with leakage-controlled pretraining (3.13; Table 2). It leads this comparison on TISP PCC (0.336 versus 0.294 for uncompressed Mamba2) and ranks second overall to externally pretrained NTv3 (3.13 versus 2.75), with roughly twice its inference speed.

Efficiency. The separate inference benchmark measures processing cost on 100 real 450K-bp inputs, independently of task adaptation. GeneZip is $1.86\times$ faster than H-Net, while U-Net is faster still (4.85 versus 6.34 seconds in total). U-Net is close to GeneZip on CMP (0.131 versus 0.133 SCC), but has lower TISP PCC (0.178 versus 0.336).

Regional-prior ablation. Different downstream tasks have different biological focuses. The promoter-focused Default (Promoter+Genic) and PromoterDistal priors both perform well on TISP (mean PCC 0.3356 and 0.3377; Appendix E.1). GeneZip lets users choose regional pretraining targets to match the biological focus of their downstream task.

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 a 0.31 perplexity increase relative to uncompressed Mamba2, preserves downstream utility across long-context genomic tasks, and achieves the best average rank across four DNALongBench downstream tasks among compression models with leakage-controlled pretraining. Without repeat-specific supervision, GeneZip exhibits emergent repeat-aware compression, allocating fewer tokens per base to TE-derived and tandem-repeat DNA than to the surrounding non-repeat sequence. GeneZip also supports 128K-context and 636M-parameter training, while its 70M model achieves a $36.1\times$ inference speedup over JanusDNA-7.7M for 450-kb inputs on a single H100 GPU. These results demonstrate high, region-aware compression with emergent repeat-aware allocation and support GeneZip as an efficient compression interface for long-context DNA modeling.

Acknowledgments and Disclosure of Funding

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, the Canada CIFAR AI Chair Program, and an NRC Collaborative R&D Project (AI4D-CORE-132). This work was also partially funded by the IVADO Fundamental Research Project grant PRF-2019-3583139727. Computational resources were provided by Mila, Calcul Québec, the Digital Research Alliance of Canada, and the National Research Council Canada.

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. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2314. URL `https://proceedings.neurips.cc/paper_files/paper/2025/hash/638fdcfbbe10b2040b0b2b0dbfb2b2c0-Abstract-Conference.html`.

[7] 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.

[8] 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.

[9] Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZbfLR9NbNF.

[10] 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.

[11] 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.

[12] 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.

[13] Daria Ledneva and Denis Kuznetsov. LDARNet: DNA adaptive representation network with learnable tokenization for genomic modeling. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 63306–63324. PMLR, 2026. URL https://proceedings.mlr.press/v306/ledneva26a.html.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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/.

[19] 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.

[20] 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.

[21] 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.

[22] Adam Frankish, Mark Diekhans, Anne-Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M. Mudge, Cristina Sisu, James Wright, Joel Armstrong, If 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.

[23] 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`.

[24] 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`.

[25] 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`.

[26] 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`.

[27] Kseniia Dudnyk, Donghong Cai, Chenlai Shi, Jian Xu, and Jian Zhou. Sequence basis of transcription initiation in the human genome. *Science*, 384(6694):eadj0116, 2024. doi: 10.1126/science.adj0116.

[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] 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.

[31] 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.

[32] 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 13.

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. For the GeneZip model, we fix the base BPT anchor to $N = 32$ bp/token across both stages. The GeneZip model focuses on genic information, with RAR objective 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 default 70M model uses unrestricted upper routing budgets; the 636M model uses ceilings of $K_{\max}^{(1)} = 30{,}000$ and $K_{\max}^{(2)} = 10{,}000$. 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 downstream adaptation uses multi-GPU parallelism because of its model size and sharding constraints; the short-context inference-only compression measurement uses one A100.

Pretraining baselines. We train comparable-size models (70M–84M) on 100B base pairs of GENCODE data with a 12.8K context. The U-Net baseline replaces GeneZip's encoder with an autoregressive U-Net [24]. H-Net-70M uses a high-compression target of 128 bp/token; H-Net-3BPT-70M provides a low-compression reference following Hwang et al. [9]. Mamba2 provides an uncompressed reference [23].

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 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) 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

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

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

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

and do not redistribute the original DNALongBench datasets or public baseline checkpoints in this work.

B Inference Efficiency

B.1 Length scaling

We measure how inference latency scales with input length, using a fixed batch size of one (Figure 3). We measure the forward-pass latency on a single NVIDIA A100 80GB GPU with BF16 inference. 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, input construction and host-to-device transfer. 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.

B.2 Inference speed

This section reports the total time to process 100 real genomic sequences of 450K bp using model-specific batch sizes. Table 2 reports processing time, and Table 4 reports the corresponding batch sizes and peak allocated GPU memory.

We use the same first 100 alternate sequences from the *Artery_Tibial* test split on one NVIDIA H100 80GB GPU with BF16, retaining the dataset's orientation, variant masking and N padding without truncation. We evaluate pretrained backbones with mean pooling and newly initialized binary classification heads, independently of task-specific adaptation. Inputs are pretokenized in pinned CPU memory.

Timing includes host-to-device transfer, mask construction, the backbone forward pass, pooling and classification; it excludes checkpoint loading, tokenization, compilation, warmup, output transfer to CPU and numerical checks. We report the median of three complete passes after warmup, synchronizing CUDA before and after each pass. Each model uses the largest batch size verified on all 100 inputs, including a final partial batch; outputs are checked against batch-1 inference with absolute and relative tolerances of 0.02.

In this inference-only benchmark, GeneZip uses routing ceilings of $(70{,}000, 70{,}000)$, and H-Net has no routing limits; these timing conditions are separate from the common downstream routing settings in Appendix C. NTv3 pads inputs to 450,048 positions, masks the added positions from pooling, and includes padding cost in timing. Peak allocated GPU memory includes model weights and is measured at each model's reported batch size.

Table 4: Batch sizes and peak GPU memory for the inference speed benchmark reported in Table 2. Measurements use one H100 80GB GPU with BF16. Each batch size is verified on all 100 inputs, with the final partial batch included in timing. Peak allocated memory includes model weights and is measured at the reported batch size. Dashes indicate unavailable measurements under this setup.

Model	Size (M)	Batch size	Peak memory (GiB)
HyenaDNA	6.6	6	41.15
Caduceus-Ph	7.7	19	75.04
Caduceus-PS	7.7	12	73.16
JanusDNA-7.7M	7.7	9	41.11
NTv3-100M-pre	100	7	56.34
Evo2-1B	1,000	—	—
Mamba2	84	5	61.34
U-Net	70	47	70.00
H-Net-70M	70	5	62.16
GeneZip	70	7	72.80

GeneZip processes the 100 inputs in 6.34 seconds, compared with 11.78 seconds for H-Net and 228.63 seconds for JanusDNA-7.7M, corresponding to $1.86\times$ and $36.1\times$ speedups. U-Net is faster at 4.85 seconds, while GeneZip achieves a better average rank across the four downstream tasks (3.13 versus 4.00; Section 4.3).

C Complete Results for DNALongBench

Task coverage. We report eQTL, ETGP, TISP, and CMP. Regulatory sequence activity prediction (RSAP) is excluded because the released data and configuration differ from the published task specification in sequence length, mouse split completeness, and training loss [3]. These discrepancies prevent a directly comparable evaluation under the published protocol.[3]

Shared comparison protocol. Stronger compression leaves fewer tokens to represent predictive information and can reduce downstream accuracy. For a fair comparison, we compare GeneZip, U-Net, and H-Net at similar validation compression around 128 BPT, using the same leakage-controlled pretraining corpus and the same DNALongBench task data, splits, and official input lengths. GeneZip achieves slightly higher compression: 137.6 BPT, compared with 128.0 for U-Net and 122.2 for H-Net (Table 2). We use the default GeneZip checkpoint from Appendix A.1 across all four tasks; alternative regional priors are evaluated in Appendix E.1. Table 2 distinguishes these models from official benchmark results and externally pretrained models.

For eQTL, ETGP, and TISP, the three models share AdamW learning rate 2×10^{-4}, weight decay 0.1, per-device batch size 1, and gradient clipping $\|g\|_2 \leq 1.0$. We freeze the compression frontend and train the subsequent backbone modules and task head. For GeneZip and H-Net, this freezes the embeddings, encoder, and router; for U-Net, it freezes the embeddings, stem, and downsampling blocks while leaving the bottleneck and upsampling blocks trainable. CMP instead trains a contact-map head on fully frozen sequence representations (Appendix C.4).

Compression measurement. Table 2 reports pretraining-validation BPT. For GeneZip (137.5858), H-Net (122.1925), and Mamba2, BPT is computed from the pretraining-validation evaluation in Table 1 as total valid input bases divided by total final-stage hard token counts. For U-Net, NTv3, HyenaDNA, both Caduceus variants, and Evo2-1B, we measure BPT on the first 3,000 validation windows using BF16 and batch size 1. Each 12,800-bp window contributes 12,799 input bases after the causal next-token shift; sequences are uppercased and no special tokens are added. We count bottleneck positions for U-Net, positions entering the first compressed transformer layer for NTv3, and final sequence-representation positions for uncompressed backbones. Fixed-downsampling padding is included in the token count but excluded from the base count: U-Net and NTv3 emit 100 tokens per window, giving 127.99 BPT. JanusDNA-7.7M's 1.0 BPT is inferred from its published single-nucleotide representation and position-wise fusion [6]; directional copies are not additional

[3]Task specification and configuration: `https://github.com/ma-compbio/DNALONGBENCH`; released data: `https://doi.org/10.7910/DVN/MNUEZR`.

Table 5: DNALongBench eQTL results across nine tissues (test AUROC; higher is better). Evaluation protocols are detailed in Appendix C.1. The best and second-best scores in each column are **bold** and underlined, respectively.

Model	eQTL Datasets and Average (AUROC)									
	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]	0.479	0.513	0.584	0.487	0.511	0.471	0.544	0.529	0.512	0.514
Caduceus-Ph[3]	0.547	0.541	0.597	0.538	0.588	0.586	0.574	0.527	0.594	0.566
Caduceus-PS[3]	0.536	0.519	0.549	0.523	0.552	0.529	0.541	0.547	0.542	0.538
External-pretraining										
JanusDNA-7.7M[6]	**0.852**	**0.769**	0.802	**0.864**	**0.914**	**0.903**	**0.846**	**0.793**	0.821	**0.840**
NTv3-100M-pre[20]	0.805	0.744	0.698	0.798	0.885	0.886	0.727	0.778	0.808	0.792
Evo2-1B[19]	0.745	0.725	0.691	0.802	0.816	0.831	0.705	0.698	0.771	0.754
Leakage-controlled-pretraining										
Mamba2	0.750	0.729	0.699	0.806	0.847	0.820	0.713	0.701	0.792	0.762
U-Net	0.845	0.764	0.724	0.837	0.892	0.876	0.830	0.715	0.776	0.807
H-Net-70M[9]	0.821	0.721	0.701	0.792	0.834	0.884	0.834	0.751	0.779	0.791
GeneZip	0.846	0.730	**0.818**	0.831	0.902	0.881	0.785	0.712	**0.849**	0.817

Note. Due to resource limitations, Evo2-1B was fine-tuned using LoRA on 4×H100 GPUs; GeneZip runs used one A100 or H100 GPU per run, and the remaining models were fine-tuned on a single A100 GPU.

sequence positions. Please note that all these BPT values describe pretraining-validation sequences; downstream compression also depends on task inputs and routing settings.

Regional-prior evaluation and routing settings. The matched-BPT regional-prior study measures the four GeneZip checkpoints and H-Net on a common 3,000-window pretraining-validation panel (Table 12). On this panel, GeneZip and H-Net have BPT values of 139.3533 and 122.4828, respectively. This is a separate measurement from the pretraining-validation statistics in Table 1.

Table 12 reports held-out test metrics. H-Net, U-Net, and all four GeneZip priors follow the common adaptation protocols and original DNALongBench sequence lengths in Appendix C; CMP uses the frozen-representation protocol in Appendix C.4. eQTL reports AUROC and AUPRC averaged over nine tissues.

For eQTL, ETGP, and TISP, H-Net and all four GeneZip priors use the same ceiling of 70,000 tokens at each routing stage, during both adaptation and validation/test evaluation. No additional routing floor is applied; content-dependent routing and the first-valid-token rule remain enabled. The ceiling limits the number of retained tokens rather than imposing a fixed token count.

C.1 eQTL prediction

We evaluate long-range variant effect modeling on the nine DNALongBench [3] eQTL classification tasks. Table 5 reports held-out test AUROC for each tissue and the unweighted mean across nine tissues.

eQTL uses 450K-bp inputs, mean pooling, three training epochs, gradient accumulation 8, and BF16. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$ with a linear learning-rate schedule and warmup ratio 0.1. Validation and checkpoint saving occur every 100 optimizer steps; the checkpoint with highest validation AUROC is evaluated on the held-out test split. eQTL inputs preserve the benchmark's orientation, variant masking, and N padding, and test predictions use the dataset-provided orientation.

Among the models with leakage-controlled pretraining in Table 5, GeneZip achieves the highest mean eQTL AUROC (0.8172), outperforming uncompressed Mamba2 across all nine tissues (Table 5).

Table 6: DNALongBench enhancer-target gene prediction results (test AUROC and AUPRC). 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-7.7M[6]	0.822	0.438
NTv3-100M-pre[20]	0.823	**0.440**
Evo2-1B[19]	0.737	0.095
Leakage-controlled-pretraining		
Mamba2[23]	0.783	0.322
U-Net	0.849	0.390
H-Net-70M	0.688	0.225
GeneZip-70M	0.823	0.375

These results show that high compression can preserve information useful for long-range variant-effect prediction.

C.2 Enhancer-target gene prediction

We assess regulatory interaction prediction on the DNALongBench enhancer-target gene prediction (ETGP) task [3], using CRISPRi_EPI_K562_hg19. The benchmark restricts candidate enhancers to within 450 kbp of the target TSS. We use the same 450K-bp inputs, mean pooling, training schedule, and validation-AUROC checkpoint selection as for eQTL (Appendix C.1).

ETGP reveals a different preference for where resolution is retained. GeneZip improves AUPRC over region-agnostic H-Net (0.3746 versus 0.2251), but uniform-downsampling U-Net is slightly stronger (0.3896; Table 6). Enhancer–target prediction links distal regulatory sequence with a target promoter, so prioritizing promoter and genic detail can leave useful enhancer context coarsely represented. The regional-prior comparison supports this interpretation: GeneZip-Regulome and GeneZip-PromoterDistal retain finer resolution in broader noncoding regions and achieve AUPRC of 0.5049 and 0.4893, respectively (Appendix E.1). These gains show why the default promoter/genic preference need not be the strongest allocation for regulatory-interaction prediction. Adjusting the prior provides a way to preserve the sequence context required by the task.

C.3 Transcription initiation signal prediction

We evaluate nucleotide-resolution transcription initiation signal prediction (TISP) on DNALong-Bench [3], predicting 10 strand-specific tracks across five assays (Table 7).

TISP uses 100K-bp inputs, a common per-base regression head, 5,000 updates, and gradient accumulation 4. Training uses a pseudo-Poisson KL loss with the ratio loss disabled. H-Net and all GeneZip priors use the shared 70,000-token stage ceilings without additional floors (Appendix C). Checkpoints are selected by the five-assay mean PCC on the validation chromosome using the official evaluation. The larger Evo2-1B model makes full fine-tuning memory-intensive, so we use LoRA to reduce its memory requirements.

On held-out chromosomes, we score the central 50 kb of overlapping 100 kb windows and compute PCC for each assay after combining both strands, then report the mean across the five assays.

GeneZip achieves a mean TISP PCC of 0.3356, exceeding H-Net (0.3032), U-Net (0.1781), and uncompressed Mamba2 (0.294; Table 7). It leads the non-expert models in Table 7 across all five assays. A plausible biological explanation is that transcription initiation depends on locally

Table 7: DNALongBench transcription initiation signal prediction across five assays [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-7.7M[6]	0.003	-0.001	0.003	0.000	0.003	0.002
NTv3-100M-pre[20]	0.220	0.148	0.149	0.170	0.284	0.194
Evo2-1B (LoRA)[19]	-0.002	-0.002	-0.004	0.000	0.000	-0.002
Leakage-controlled-pretraining						
Mamba2[23]	0.336	0.248	0.240	0.264	0.382	0.294
U-Net[24]	0.159	0.142	0.143	0.168	0.279	0.178
H-Net-70M[9]	0.341	0.261	0.249	0.271	0.393	0.303
GeneZip-70M	0.392	0.301	0.286	0.288	0.411	0.336

organized promoter features whose positions influence where transcription begins, as demonstrated by Puffin [27]. GeneZip's regional prior favors finer token resolution around promoters, which may help preserve these position-sensitive features during compression. This interpretation is consistent with the lower TISP performance of GeneZip-Intergenic (a control model that reverses GeneZip's default promoter/genic preference), which shifts resolution away from promoters (Appendix D.1).

C.4 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 [19] with precomputed coverage-aware pooled bin embeddings from the frozen backbone; NTv3-100M-pre; JanusDNA-7.7M [6]; Mamba2 without compression; U-Net with uniform compression at 128 bp per token; H-Net-70M; and GeneZip.

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.

GeneZip achieves the highest mean SCC in the leakage-controlled group (0.1332), exceeding U-Net (0.1314), H-Net (0.1182), and uncompressed Mamba2 (0.1108; Table 8). This result shows that its highly compressed representations retain information useful for long-range contact prediction. Externally pretrained JanusDNA has the highest mean SCC (0.1590), while NTv3 reaches 0.1361.

Table 8: DNALongBench contact map prediction across five cell lines [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-7.7M	**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-70M	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	0.1324	0.1840	0.1338	0.1867	0.1071	0.1600	0.1310	0.1804	0.1618	0.1843	0.1332	0.1791

GeneZip approaches NTv3's CMP performance with leakage-controlled pretraining and requires about half its inference time in the separate 450K-bp benchmark (Table 2).

D Compression Analysis

Using checkpoints pretrained on the same leakage-controlled human corpus (Appendix F), we examine regional budget consistency and preference transfer (Section D.1), repeat-aware compression without repeat supervision (Section D.2), and routing at representative loci (Section D.3). The case studies connect aggregate allocation patterns with local genomic features.

D.1 Regional Allocation and Transfer

Research questions. GeneZip uses region-specific targets to guide compression during training, but whether the learned compressor preserves these budgets and allocation preferences on unseen sequences remains to be established. We therefore evaluate two research questions. **RQ1 (Budget consistency)** asks whether compression remains aligned with its training-time budget targets. **RQ2 (Regional preference transfer)** asks whether the regional prior's allocation preference is preserved without annotations at inference.

We compare encoder-only checkpoints of H-Net-70M, GeneZip, GeneZip-Intergenic, and GeneZip-128K on 3,094 held-out human and 2,368 mouse windows of 128 Kbp.[4] Within each species, all checkpoints use the same windows and ambiguity filter. We obtain hard boundary masks with reverse-complement averaging and measure BPT at stage 2.

Metrics. For RQ1 (budget consistency), *observed BPT* describes overall compression as input bases divided by output tokens; larger values mean coarser resolution. To assess whether this compression matches the training targets, *BPT-ratio* divides observed BPT by the *expected BPT* for the input's regional composition. It is equivalently the expected token count, obtained by summing regional base counts divided by their target BPTs (Eq. 5), divided by the observed count; values closer to one indicate closer total-budget agreement.

Total-budget agreement can conceal offsetting regional deviations, so we also measure regional target agreement with *MicroErr*. Each token is fractionally assigned to regions according to span overlap, and *regional BPT* divides a region's base count by its fractional token count. MicroErr is the base-count-weighted mean absolute natural log of regional observed-to-target BPT ratios; lower values indicate smaller regional discrepancies, with zero indicating exact agreement.

For RQ2 (regional preference transfer), we ask which regions retain relatively finer resolution. BPT describes average compression within a sequence or region; to compare allocation preferences across

[4]Human windows tile held-out chromosomes `chr8/chr9/chr10`; mouse windows tile a fixed-seed block of `chr11/chr19/chr10` from the GENCODE-derived mouse genome. Both use contiguous, non-overlapping windows and the standard GENCODE preprocessing ambiguity filter (`max_n_frac`=0.05). Of 3,262 human and 2,456 mouse windows, 168 and 88 high-`N` or assembly-gap windows are excluded, respectively.

Table 9: 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. Token Density is normalized by the sequence-wide average; values above one indicate denser-than-average token allocation.

Species	Model	Observed BPT	BPT-ratio	PPL ↓	MicroErr ↓	Intron + intergenic Token Density	Promoter Token Density	Genic Token Density
Human	H-Net-70M	124.4 ± 4.4	0.972 ± 0.034	2.8699	-	1.001 ± 0.009	0.996 ± 0.138	0.970 ± 0.232
	GeneZip-Intergenic	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	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-70M	120.7 ± 5.2	0.943 ± 0.040	3.5559	-	1.005 ± 0.011	0.978 ± 0.178	0.912 ± 0.226
	GeneZip-Intergenic	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	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

regions, we report *Token Density*, defined as each region group's tokens per base normalized by the sequence-wide average. It is the group's share of fractionally assigned tokens divided by its share of bases: values above one indicate denser allocation, and values below one indicate sparser allocation. We compare promoter, genic (CDS/UTR/exon), and intron plus intergenic (intron/NIG/DIG) groups to test whether the prior's preferred allocation direction persists.

To relate these compression measurements to reconstruction quality, we also report *base-level PPL*, the exponential of mean next-base negative log-likelihood; lower values indicate better reconstruction.

RQ1: Budget consistency. GeneZip and GeneZip-128K retain strong compression on unseen human and mouse sequences, with observed BPT of 162.7–198.8 (Table 9). Their mean BPT-ratios range from 0.919 to 1.106, indicating that total compression budgets remain close to the targets after transfer. GeneZip-Intergenic shows larger total-budget deviations, with BPT-ratios of 1.397 in human and 1.538 in mouse. Total-budget adherence therefore transfers more closely for the default promoter/genic prior than for the reversed prior. At the regional level, GeneZip-128K improves target agreement relative to GeneZip: MicroErr decreases from 0.3372 to 0.2676 in human and from 0.4189 to 0.2566 in mouse.

RQ2: Regional preference transfer. Token Density reveals distinct allocation preferences despite similar reconstruction PPL within each species (Table 9). H-Net-70M allocates tokens to promoters roughly in proportion to their sequence coverage, with Token Density near one. GeneZip instead gives promoters about twice its sequence-average token allocation per base (Token Density ≈ 2), retaining finer resolution there and in genic regions; GeneZip-128K preserves this preference at longer context. GeneZip-Intergenic is a comparison model designed to reverse GeneZip's default regional preference (Appendix E.1). As expected, it yields promoter Token Density of 0.6–0.7 and genic Token Density below one, shifting relative resolution toward intron/intergenic sequence. Both allocation preferences persist on held-out human and mouse sequences without annotations at inference. These results show that the regional prior can be adjusted to users' preferences for where to retain finer resolution. The downstream effects of these allocation preferences are examined in Appendix E.1.

D.2 Repeat-aware Compression

We compare GeneZip and H-Net on two matched samples of approximately 10 Mb: 78 non-overlapping 128K-bp windows from held-out human chromosomes `chr8/chr9/chr10` and 78 mouse evaluation windows. RepeatMasker TE classes (DNA transposon, LINE, LTR, and SINE) and TRF masks define the evaluated repeat positions; matched non-repeat positions are drawn from the same full-window context. Tandem-repeat statistics retain windows with at least 1 kb of TRF-masked sequence and 10 kb of non-TRF sequence, leaving 70 human and 78 mouse windows. A repeat/non-repeat PPL ratio below one indicates easier reconstruction, while a BPT ratio above one indicates stronger local compression.

Stronger repeat compression at similar PPL. GeneZip compresses every evaluated repeat subset more strongly than its matched non-repeat background, and more strongly relative to that background

Table 10: **Repeat predictability and compression allocation: GeneZip versus H-Net.** Human/mouse measurements underlying Figure 4, using panels of 78 full 128K-bp windows per species; TR statistics use the 70 human and 78 mouse windows passing the support filter in Appendix D.2. PPL columns report mean repeat-position PPL / matched-background PPL / their ratio, averaged per window; BPT columns report mean repeat/non-repeat ratios. PPL ratio below 1 indicates easier reconstruction and BPT ratio above 1 indicates stronger local compression. Despite nearly identical PPL, GeneZip has higher BPT ratios than H-Net in every subset shown.

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

than H-Net does (Table 10). Its repeat/non-repeat BPT ratios exceed one and are higher than H-Net's in all ten human/mouse comparisons, despite nearly identical repeat-position and matched-background PPL. The strongest TE-class contrast in human is SINE, with BPT ratios of 2.67 for GeneZip and 1.24 for H-Net. For tandem repeats, GeneZip's ratios exceed one in both species, while H-Net's remain below one. Most repeat subsets also have lower PPL than the background; mouse LTR is an exception, with PPL ratios slightly above one for both models. Thus, similar reconstruction quality can coexist with substantially different token allocation: GeneZip more consistently assigns fewer tokens to repeats relative to their non-repeat background.

Transfer across nine species. To test whether this repeat-compression preference extends beyond the human training genome, we evaluate the human-trained GeneZip and H-Net checkpoints without further training on a separate RefSeq panel spanning mouse, rat, dog, cow, chicken, frog, zebrafish, Drosophila, and Arabidopsis. Each species contributes 16 sequence-deduplicated, non-overlapping 12.8K-bp windows with at most 5% ambiguous bases.

Using identical windows and TRF masks, GeneZip has higher TR/non-TR BPT ratios in all nine species, with an equal-weight species mean of 1.995 versus 1.406 for H-Net (Table 11). GeneZip's ratio exceeds one in all nine species, and H-Net's in eight: H-Net also compresses TR more than the background in most species of this separate panel, but GeneZip does so more strongly. Both models have TR PPL ratios below one in every species, while their PPL-ratio ordering varies. Together, the human/mouse and nine-species comparisons show that GeneZip learns stronger repeat-aware compression than H-Net without repeat-specific supervision, and that this allocation preference extends beyond the human training genome.

D.3 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 [22]. Figure 5 overlays region labels (promoter/CDS/UTR/exon/intron/near-intergenic (NIG)) with per-base boundary probabilities produced by GeneZip ($\mathrm{BPT} = 137.6$) and an H-Net-70M baseline ($\mathrm{BPT} = 122.2$) [9]. The plotted probabilities describe routing preferences; the final retained tokens also depend on thresholding and bounded routing.

The two loci illustrate complementary allocation patterns. In the promoter-dense locus, GeneZip raises boundary probability near several promoter and exonic features while maintaining lower values between them. In the intron-dominated locus, a long interval of low boundary probability coexists with localized peaks near annotated landmarks. H-Net has a more diffuse background with isolated spikes. These examples illustrate how the aggregate regional Token Density patterns in Appendix D.1 can appear locally along a sequence.

Table 11: **Tandem-repeat compression across nine species.** Human-trained models are evaluated without further training on the same 16 full 12.8K-bp windows per species. n counts windows containing TR; PPL and BPT ratios compare TR with non-TR positions and are averaged over eligible windows. GeneZip has higher BPT ratios than H-Net in all nine species, while PPL-ratio ordering varies.

Species	n	GeneZip PPL ratio	H-Net PPL ratio	GeneZip BPT ratio	H-Net BPT ratio
Mouse	16	0.539	0.538	1.571	0.827
Rat	16	0.621	0.621	2.091	1.783
Dog	16	0.625	0.629	1.753	1.270
Cow	16	0.923	0.782	1.855	1.488
Chicken	16	0.647	0.643	2.137	1.385
Frog	16	0.830	0.916	2.733	1.719
Zebrafish	16	0.635	0.658	1.986	1.167
Drosophila	15	0.928	0.781	1.645	1.508
Arabidopsis	11	0.833	0.852	2.186	1.510

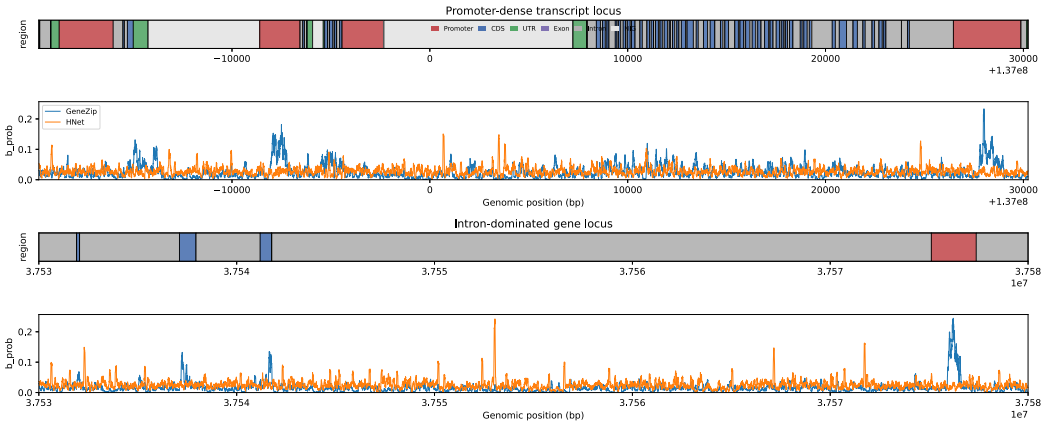


Figure 5: **Case studies of region-adaptive compression.** Region labels are induced from GENCODE gene models [22] (promoter/CDS/UTR/exon/intron/near-intergenic (NIG)), and the curves plot per-base boundary probability for H-Net [9] and GeneZip. We highlight two loci on chr9: a promoter-dense transcript locus (chr9:136,980,237–137,030,237; top) and an intron-dominated gene locus (chr9:37,530,000–37,580,000; bottom). GeneZip shows localized boundary-probability peaks near several promoter and exonic features, with a lower background across long intronic spans.

E Ablation Studies

E.1 Regional-prior comparison

Our default regional prior favors promoter and genic resolution because coding exons, UTRs, and promoters are enriched for evolutionarily constrained bases, reflecting the importance of preserving sequence detail in these regions [7]. We encode this preference through smaller μ_r for promoters and genic elements and larger values for intronic and intergenic sequence. Gene regulation depends on both promoters and distal regulatory elements [29, 30], motivating comparisons with priors that redirect resolution toward broader noncoding and intergenic regions. These comparisons test how the location of retained resolution affects downstream prediction.

We compare four regional allocation priors at approximately matched pretraining-validation BPT, using one pretrained checkpoint per prior across all four downstream tasks. In the RAR objective, $N_r^\star = N\mu_r$ specifies the target BPT for region r; smaller μ_r preserves finer resolution. In the order (promoter, CDS, UTR, exon, intron, NIG, DIG), the multipliers are $(1, 1, 2, 2, 8, 8, 16)$ for GeneZip, $(1, 16, 2, 4, 2, 2, 4)$ for GeneZip-Regulome, $(1, 16, 8, 8, 2, 2, 4)$ for GeneZip-PromoterDistal, and $(32, 16, 8, 4, 2, 4, 1)$ for GeneZip-Intergenic. GeneZip prioritizes promoter/CDS resolution, GeneZip-Regulome emphasizes promoter and broader noncoding regulatory context, GeneZip-PromoterDistal

Table 12: Regional-prior comparison across four tasks at approximately matched pretraining-validation BPT. Each GeneZip row uses one pretrained checkpoint across all tasks. GeneZip-Regulome and GeneZip-PromoterDistal use $N = 64$; GeneZip and GeneZip-Intergenic use $N = 32$. BPT is measured on the same 3,000 pretraining-validation windows; U-Net uses its architectural count. All scores are held-out test metrics. H-Net and all GeneZip priors use a common 70,000-token ceiling per stage without additional routing floors for eQTL, ETGP, and TISP. eQTL reports mean AUROC/AUPRC over nine tissues, ETGP reports AUROC/AUPRC, CMP reports mean SCC/Corr over five cell lines, and TISP reports mean PCC over five assays. Best scores within this table are **bold**.

Model / prior	Validation BPT	eQTL AUROC/AUPRC	ETGP AUROC/AUPRC	CMP SCC/Corr	TISP PCC
U-Net	127.99	0.8067 / 0.2019	**0.8488** / 0.3896	0.1314 / 0.1762	0.1781
H-Net-70M	122.48	0.7909 / 0.1609	0.6881 / 0.2251	0.1182 / 0.1607	0.3032
GeneZip	139.35	0.8172 / 0.1995	0.8229 / 0.3746	**0.1332 / 0.1791**	0.3356
GeneZip-Regulome ($N = 64$)	118.15	0.8229 / 0.1978	0.8290 / **0.5049**	0.1013 / 0.1482	0.3278
GeneZip-PromoterDistal ($N = 64$)	122.79	**0.8241** / 0.1847	0.8087 / 0.4893	0.0932 / 0.1416	**0.3377**
GeneZip-Intergenic ($N = 32$)	115.95	0.7865 / **0.2058**	0.8185 / 0.4230	0.1280 / 0.1735	0.2654

further emphasizes promoter, intron and near-intergenic sequence, and GeneZip-Intergenic prioritizes distal intergenic sequence as a counter-prior to the default promoter/genic preference. GeneZip-Regulome and GeneZip-PromoterDistal use $N = 64$; GeneZip and GeneZip-Intergenic use $N = 32$. U-Net and H-Net provide uniform-downsampling and region-agnostic routing baselines, respectively.

Evaluation follows the shared adaptation protocols and common-panel BPT measurement described in Appendix C.

Task-dependent biological focus. Different downstream tasks emphasize different biological features, motivating different regional pretraining targets. For TISP, the two promoter-focused priors, Default (Promoter+Genic) and PromoterDistal, both perform well, with mean PCC of 0.3356 and 0.3377, respectively (Table 12). The default prior leads on FC, GC, and PC, whereas PromoterDistal leads on EC and ER. Their shared emphasis on promoter resolution is consistent with the position-sensitive role of promoter features in transcription initiation [27].

Tasks involving distal regulatory context favor different allocations. Regulome achieves the highest ETGP AUPRC among the GeneZip priors (0.5049), while PromoterDistal has the highest eQTL AUROC (0.8241). Both exceed the default prior on these tasks, but have lower CMP SCC (0.1013 and 0.0932 versus 0.1332 for the default). These results illustrate GeneZip's flexibility: users can choose regional pretraining targets to preserve resolution in regions relevant to the biological focus of their downstream task.

Biological interpretation of the reversed prior. GeneZip-Intergenic provides a direct contrast to the default promoter/genic preference: its learned allocation shifts resolution away from promoters and genic regions toward intron/intergenic sequence (Appendix D.1). On the common pretraining-validation panel, it uses more tokens per unit sequence than GeneZip, with BPT of 115.95 versus 139.35. Yet its TISP PCC falls from 0.3356 to 0.2654, a 20.9% reduction, with lower performance across all five assays. The greater overall token density therefore does not translate into better transcription-initiation prediction when promoter-associated resolution is reduced.

This result is consistent with the biological importance of preserving sequence detail in promoters and genic elements, where evolutionarily constrained bases are enriched [7]. Such constraint reflects purifying selection against changes that disrupt function. For transcription initiation, promoter motifs exert position-specific effects [27], providing a plausible explanation for the lower TISP performance when resolution is shifted away from promoters. The GeneZip-Intergenic comparison thus suggests that retaining more tokens elsewhere cannot necessarily compensate for coarser representation of these functionally important sequences.

F Training Data Details

We construct the human pretraining corpora from the GRCh38 primary assembly together with the GENCODE v49 *basic* gene annotation [22].

Table 13: 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%)

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, 31], 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 determines the units on which a DNA model operates. Viewed together with the embedding step, it serves the same role as GeneZip's encoder in Eq. 1: both map a raw nucleotide sequence to input representations for the token-mixing backbone. In this broad sense, GeneZip can be understood as end-to-end learned latent tokenization.

However, GeneZip focuses on extreme sequence-length compression for long-context modeling, whereas existing DNA tokenization methods primarily emphasize useful sequence representations. From a compression perspective, the two representative methods in Table 14 achieve modest reductions in sequence length. BPE [32], whose merge rules are fixed during model training, yields 1.4 bp/token on the eQTL Artery Tibial test set. MxDNA [10], which learns tokenization jointly with the model, yields 3.0 bp/token on the same sequences. Even at this resolution, adapting the released MxDNA Transformer to 450K-bp eQTL inputs exceeds the available GPU memory. GeneZip supports this long-context evaluation and achieves a mean eQTL AUROC of 0.8172, compared with 0.7544 for BPE.

H Limitations and Future Work

GeneZip is currently pretrained on human genomic data. The full-chromosome human/mouse compressor analysis in Table 9 provides inference-time evidence that the learned routing policy transfers across held-out chromosomes and to another mammalian genome while still concentrating

Table 14: Comparison of GeneZip with tokenization baselines on long-context downstream tasks. BPE and MxDNA BPT is evaluated on the eQTL Artery Tibial test set and computed as total bp divided by total tokens; the dash indicates that GeneZip has no reported BPT measurement on this panel. GeneZip task scores are from Tables 5 and 6. BPE uses the same backbone and GENCODE training data as GeneZip; MxDNA uses the authors' released Transformer checkpoint and thus differs in backbone architecture. GeneZip follows the adaptation protocols in Appendix C. OOM denotes an out-of-memory failure on 450K-bp inputs.

Model	**BPT**	**eQTL AUROC**	**ETGP AUPRC**
BPE	1.3947	0.7544	0.2540
MxDNA	3.0283	OOM	OOM
GeneZip	—	0.8172	0.3746

token budget according to the selected biological prior. Extending pretraining to broader multi-species resources with harmonized region annotations, together with downstream benchmarking across species, is an important direction for future work.

GeneZip builds on the underlying H-Net routing architecture [9]. 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, 20]. We do not anticipate direct negative societal impact from the present methodological contribution.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sections 4.1, 4.2, and 4.3 support the compression, repeat-allocation, efficiency, and downstream claims. The downstream comparison uses a fixed regional prior and distinguishes shared-pretraining baselines from externally pretrained models; adaptation differences are described in Appendix C.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix H discusses human-genome pretraining, dependence on static annotations, and reliance on H-Net routing. Appendices C and E.1 describe the scope of the comparisons and differences in downstream adaptation.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical guarantees or theorem-based results. Section 3.2 defines the compression objective and routing mechanism.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The architecture and objectives are specified in Section 3.2; pretraining, data construction, and downstream protocols are described in Appendices A.1, F, and C. The public code at `https://github.com/AndyJZhao/GeneZip` provides GeneZip implementations and reproduction instructions for pretraining and the four DNALongBench tasks.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: GeneZip code and reproduction instructions for the main experiments—pretraining and the four DNALongBench tasks (CMP, eQTL, ETGP, and TISP)—are publicly available at `https://github.com/AndyJZhao/GeneZip`. The repository documents the environment, pretrained checkpoints, and commands for training and evaluation. The source datasets and data construction procedures are described in Appendices A.2 and F.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendices A.1 and F specify pretraining hyperparameters, corpus construction, and leakage exclusions. Appendix C gives task-specific training, evaluation, and checkpoint-selection settings, and Appendix E.1 states the settings and comparability limits of the regional-prior ablation.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Table 9 reports variation across held-out sequence windows, not across training seeds. The downstream results are point estimates without multi-seed confidence intervals or significance tests; score differences and average ranks do not establish statistical significance.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.1 describes accelerator types and pretraining budgets. Appendix B.2 reports inference timing, batch sizes, and peak GPU memory, and Appendix C specifies downstream hardware and training schedules.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public genomic reference resources and existing benchmark datasets for methodological research, with sources and terms described in Appendix A.2. It does not recruit participants or collect new human-subject data.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper discusses the benefit of reducing the computational cost of long-context genomic modeling. Appendix H states that no direct negative societal impact is anticipated, but does not provide a detailed assessment of potential negative impacts or misuse.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The contribution is a sequence-compression method evaluated on public genomic references and prediction benchmarks, rather than a system for designing or experimentally validating biological sequences. The release does not introduce a new participant-level genomic dataset; no application-specific release safeguards are described.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The manuscript cites the datasets, annotations, codebases, and baseline models used in the experiments. Appendix A.2 describes their licenses or source terms, and the public code repository includes third-party notices.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The GeneZip code release at `https://github.com/AndyJZhao/GeneZip` includes an Apache-2.0 license, third-party notices, dependency specifications, and instructions for pretraining and downstream evaluation. Its README documents checkpoint variants, regional multipliers, and reproduction commands for the four DNALongBench tasks.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The study uses existing public genomic reference resources and benchmark datasets. It does not involve crowdsourcing, participant recruitment, or new human-subject experiments.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The study does not recruit participants or collect new human-subject data. The experiments use existing public reference resources and benchmark datasets under their source terms.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core research methodology. The compression method and its training objectives are specified in Section 3.2.