JEPA-DNA: GROUNDING GENOMIC FOUNDATION MODELS THROUGH JOINT-EMBEDDING PREDICTIVE ARCHITECTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Genomic Foundation Models (GFMs) typically rely on Masked Language Modeling (MLM) or Next-Token Prediction (NTP) to learn the "Laws of Nature". While effective at capturing local syntax, these generative paradigms prioritize token-level reconstruction over high-level functional context. We introduce JEPA-DNA, a model-agnostic continual training framework that integrates a Joint-Embedding Predictive Architecture (JEPA) with traditional generative objectives. By supervising global sequence embeddings in a latent space, JEPA-DNA forces models to predict the functional representations of masked genomic segments, shifting the learning signal from token recovery to semantic alignment. We evaluate JEPA-DNA on 17 diverse genomic benchmark tasks, demonstrating consistent gains in linear probing and zero-shot performance regardless of the underlying GFM architecture or generative objective, and establishing a new state-of-the-art for GFMs under the linear-probing protocol. Through extensive ablation studies, we further characterize the synergistic interplay between generative and latent objectives. Our code is provided in the supplementary material.

1 INTRODUCTION

Genomic Foundation Models (GFMs) have significantly advanced our ability to model the complex interactions within DNA sequences (Benegas et al., 2025c). Architectures such as DNABERT-2 (Zhou et al., 2024), Nucleotide Transformer (NT) (Dalla-Torre et al., 2025), HyenaDNA (Nguyen et al., 2023), Evo (Nguyen et al., 2024) and Caduceus (Schiff et al., 2024) adapt Large Language Models (LLMs) to genomic data, utilizing context windows from several kilobases to megabase scales over backbones that range from Transformers to long convolutions and bi-directional Mamba layers. Typically, these models rely on generative self-supervised objectives, such as Masked Language Modeling (MLM) or Next Token Prediction (NTP), to learn representational spaces.

While effective at identifying local motifs and sequence patterns (Zhou et al., 2024; Dalla-Torre et al., 2025), generative objectives focus heavily on token-level reconstruction. Motivated by joint-embedding predictive architectures, which suggest that reconstruction in input space can prioritize low-level details over more abstract semantic structure (Assran et al., 2023), we propose a framework designed to capture the high-level regulatory logic and long-range interactions that govern genomic function.

To bridge the gap between genomic syntax and biological semantics, we introduce JEPA-DNA, a novel framework that integrates the Joint-Embedding Predictive Architecture (JEPA) (LeCun et al., 2022) into genomic training. In JEPA, the predictor must recover the latent representation of a masked span from the surrounding context, which can only be satisfied by encoding sequence-level regularities rather than local token statistics. By coupling token-level recovery with latent-space prediction, JEPA-DNA encourages the learning of abstract, functional features invariant to low-level sequence noise.

We evaluate JEPA-DNA across 17 genomic benchmark tasks, utilizing linear probing and zero-shot protocols to isolate representational quality (Larey et al., 2026). Our approach is uniquely modular, serving as a continual pre-training phase to refine existing GFMs across disparate architectures and objectives. Empirical results demonstrate that latent prediction consistently improves performance,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

establishing a new state-of-the-art for GFMs across supervised linear probing tasks. Extensive ablation studies confirm that performance gains are driven by JEPA-DNA's core components rather than simply additional training.

In summary, our contributions are as follows:

- We introduce a novel continual training framework for GFMs that shifts the learning objective from literal token reconstruction to latent feature alignment via a Joint-Embedding Predictive Architecture. To the best of our knowledge, JEPA-DNA is the first to adapt Joint-Embedding Predictive Architectures to high-resolution, multiscale genomic sequences.
- We evaluate JEPA-DNA on 17 diverse genomic benchmark tasks, demonstrating consistent gains in linear probing and zero-shot performance across leading GFM architectures (DNABERT-2 (Zhou et al., 2024), NTv3 (Boshar et al., 2025), and HyenaDNA (Nguyen et al., 2023)), tokenization methods, and generative objectives (MLM and NTP).
- We establish a new state-of-the-art under our frozen linear probing settings, across supervised tasks, empirically showing that JEPA-DNA learns more linearly-separable and biologically relevant features than standard generative baselines.

2 Related Work

2.1 Pre-training Paradigms of Genomic Foundation Models

The development of GFMs has been largely inspired by the success of LLMs in Natural Language Processing. Early iterations, such as DNABERT (Ji et al., 2021), adapted the BERT architecture (Devlin et al., 2019) replacing its original WordPiece subword tokenizer with k-mer tokenization to capture bidirectional context within the genome. More recently, DNABERT-2 (Zhou et al., 2024) and the Nucleotide Transformer (Dalla-Torre et al., 2025) expanded this scale by training on diverse multi-species datasets, demonstrating that larger context windows and higher parameter counts can improve "out-of-the-box" performance on downstream tasks like promoter prediction and variant effect prediction.

Beyond Transformer-based architectures, recent advancements have focused on overcoming the quadratic scaling of self-attention to model longer genomic dependencies. HyenaDNA (Nguyen et al., 2023) utilizes the Hyena operator to process sequences at single-nucleotide resolution across long contexts. Caduceus (Schiff et al., 2024) replaces attention with bi-directional Mamba (state-space) layers. Similarly, Evo2 (Brixi et al., 2026) leverages a StripedHyena 2 backbone to push the context size to megabase scales. In parallel, Nucleotide Transformer v3 (NTv3) (Boshar et al., 2025) extends this line of work by wrapping a Transformer backbone within a U-Net–style architecture, enabling efficient downsampling and upsampling across megabase-scale genomic windows while maintaining single-nucleotide-resolution predictions. Despite these structural innovations, nearly all current GFMs rely on token-level reconstruction objectives, namely MLM and NTP. While these methods are effective for learning the "syntax" of DNA, they often fail to capture the structural "logic" or functional state of the genome (Benegas et al., 2023).

Alternative self-supervised strategies have sought to move beyond literal token reconstruction by incorporating evolutionary or contrastive constraints. Models such as GPN-MSA (Benegas et al., 2025a) leverage multi-sequence alignment (MSA) to learn from cross-species conservation patterns, identifying functional constraints through substitution probabilities. In parallel, contrastive learning frameworks like DNASimCLR (Yang et al., 2024) employ data augmentations to learn representations by minimizing the distance between similar sequence "views" in latent space. However, these paradigms involve significant trade-offs: MSA-based methods require computationally expensive preprocessing and are limited by the availability of high-quality alignments, while contrastive methods often necessitate non-trivial augmentations or costly negative mining. In contrast, JEPA-DNA captures functional invariants directly from the raw sequence.

2.2 Joint-Embedding Predictive Architecture (JEPA)

In computer vision, the Joint-Embedding Predictive Architecture (JEPA) has emerged as a powerful alternative to generative modeling. In this paradigm, a model is tasked with predicting the latent

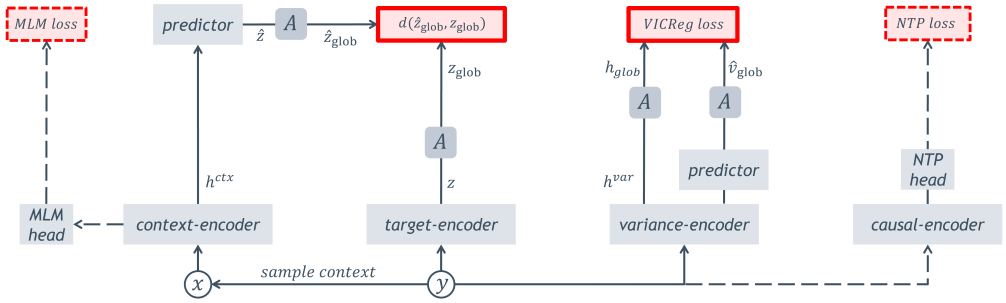

Figure 1: The JEPA-DNA Architecture.

representation of a masked segment rather than its literal pixels (Assran et al., 2023). This shift from signal reconstruction to representation prediction allows the model to ignore unpredictable, low-level details and focus on semantically rich features. This concept was recently extended to the language domain with LLM-JEPA (Huang et al., 2026), which proposes a first step towards coupling token-level recovery with sequence-level latent grounding for natural language tasks. Within the biological domain, the JEPA framework was recently introduced by GeneJEPA for single-cell transcriptomics (Litman et al., 2025). However, GeneJEPA operates on gene-expression vectors, designed to learn and reason about gene-gene relationships within a cell. Unlike transcriptomic models that process tabular gene sets, our framework must handle the multiscale nature of DNA, where functional meaning is encoded through sequential motifs and long-range dependencies.

3 Method

The JEPA-DNA framework treats the genome not merely as a sequence of tokens, but as a structured signal with both local syntax and global semantics. We achieve this by augmenting a standard GFM backbone $\mathcal{E}$ with a Joint-Embedding Predictive Architecture (JEPA) branch, enabling a multi-objective learning process.

3.1 Architecture Components

The JEPA-DNA architecture is illustrated in Fig. 1. Its core elements are as follows:

- **Context-encoder ($\mathcal{E}_\theta$):** A GFM backbone that processes the masked input sequence $\mathbf{x}$ and produces a contextual latent sequence $\mathbf{h}^{ctx} = \mathcal{E}_\theta(\mathbf{x})$.
- **Target-encoder ($\mathcal{E}_{\bar{\theta}}$):** A structural duplicate of the context encoder, whose weights $\bar{\theta}$ are updated via an exponential moving average (EMA) of θ. This encoder processes the corresponding unmasked target sequence $\mathbf{y}$ to yield a latent target sequence $\mathbf{z} = \mathcal{E}_{\bar{\theta}}(\mathbf{y})$.
- **Predictor head ($\mathcal{P}_\phi$):** A network designed to map context representations into the target latent space. It takes the contextual sequence $\mathbf{h}^{ctx}$ as input and outputs predicted target latents $\hat{\mathbf{z}} = \mathcal{P}_\phi(\mathbf{h}^{ctx})$, thereby learning to align context-derived representations with the target encoder's latent sequence.
- **Sequence aggregator ($\mathcal{A}$):** An aggregation operator that maps latent sequence embeddings to a single global sequence representation, instantiated according to the backbone's native behavior (e.g., $[CLS]$ token embedding, last token embedding, or mean pooling). Given a target latent sequence $\mathbf{z}$, it produces a global target representation $\mathbf{z}_{\text{glob}} = \mathcal{A}(\mathbf{z})$, and when applied to the predictor outputs $\hat{\mathbf{z}}$ it yields the corresponding predicted global latent representation $\hat{\mathbf{z}}_{\text{glob}} = \mathcal{A}(\hat{\mathbf{z}})$.

In addition, we introduce a **causal-encoder** for auto-regressive backbones (as detailed in Section 3.3.1) and an additional **variance-encoder** to address mode collapse (as detailed in Section 3.3.3).

3.2 Masking and Re-masking Strategy

The JEPA-DNA framework relies on a dual-masking process, to ensure the predictor does not see the masked content.

Initial Masking. Given an input sequence $\mathbf{y}$, a subset of tokens is replaced by special $[MASK]$ tokens, producing $\mathbf{x}$. This masked sequence is subsequently processed by the context encoder $\mathcal{E}_\theta$ to yield latent representations $\mathbf{h}^{ctx}$. In contrast to standard MLM protocols, which typically mask random, independent tokens (approximately 15%), our approach adopts a span-based masking strategy. We sample multiple contiguous target regions rather than individual tokens, resulting in a higher aggregate masking ratio (typically exceeding 20%), in alignment with the JEPA context encoder masking protocol in vision (Assran et al., 2023). Furthermore, we introduce a masking scheduler that gradually increases both the masking ratio and the number of masked spans over the course of training. Early in training, the model is exposed to relatively mild corruption patterns, while later stages employ more aggressive masking configurations, progressively challenging the model to infer longer-range dependencies from increasingly limited context. In particular, this masking configuration is propagated to both the JEPA loss and the standard MLM loss, thereby enforcing a uniformly more challenging reconstruction objective.

Re-masking Strategy. To prevent the predictor from having a trivial mapping to the targets, we introduce a re-masking step, where $\mathbf{h}^{ctx}$, the outputs of $\mathcal{E}_\theta$, at the masked positions are replaced by a learnable latent $[MASK]$ embedding. Note that the predictor head receives the full sequence of context encodings. By conditioning the predictor on the entire encoded sequence rather than a single vector, the model can leverage both localized spatial information and the global sequence summary to accurately reconstruct the target latent $\hat{\mathbf{z}}_{\text{glob}}$.

3.3 Multi-Objective Continual training

JEPA-DNA employs a two-stage continual training procedure starting from a GFM backbone previously pre-trained on large-scale genomic data using its native objective (MLM or NTP). During the first stage, the backbone encoder remains frozen while only the attached predictor branch is optimized. In the second stage, the encoder and predictor are updated jointly, refining the model under the combined JEPA-DNA objective.

The model is optimized by minimizing a composite loss function that balances language modeling, latent prediction, and embedding diversity.

3.3.1 LLM Loss ($\mathcal{L}_{llm}$)

To maintain token-level precision, we retain the standard LLM objective (MLM or NTP). For a masked sequence with indices $\mathcal{M}$, the loss is:

$$\mathcal{L}_{llm} = -\sum_{i \in \mathcal{M}} \log P(t_i \mid h_i) \tag{1}$$

where $P(t_i)$ denotes the predicted probability of the genomic token t_i at position i given the contextual hidden state h_i. The set of indices $\mathcal{M}$ is derived from the masking strategy in the MLM setting or from the causal autoregressive formulation in the NTP setting. In the latter case, the objective is computed using an additional forward pass over the full sequence through a causal-encoder that shares its weights with the context encoder.

3.3.2 Latent Predictive Loss ($\mathcal{L}_{jepa}$)

The JEPA objective grounds the model by forcing the global representation $\mathbf{h}_{\text{glob}}$ to capture functional semantics. The predictor's global head $\hat{\mathbf{z}}_{\text{glob}}$ attempts to match the embedding produced by the target encoder $\mathbf{z}_{\text{glob}}$ in the latent space.

We define this loss using cosine similarity:

$$\mathcal{L}jepa = 1 - \frac{\hat{\mathbf{z}}_{\text{glob}} \cdot \mathbf{z}_{glob}}{||\hat{\mathbf{z}}_{\text{glob}}||_2 \cdot ||\mathbf{z}_{glob}||_2} \tag{2}$$

Minimizing this objective encourages the predicted representation of the context encoder to align with the functional embedding of the target encoder.

3.3.3 VARIANCE AND COVARIANCE REGULARIZATION

To prevent the "collapse" problem common in non-contrastive methods (where the model outputs a constant vector), we utilize variance and covariance constraints on the latent vectors $\mathbf{h}_{\text{glob}}, \hat{\mathbf{v}}_{\text{glob}} \in \mathbb{R}^{B\times d}$ across a batch of size B and hidden dimension d, following the VICReg framework (Bardes et al., 2021). These vectors are obtained via a separate forward pass through a deterministic variance encoder (without dropouts or masking) that produces $\mathbf{h}^{\text{var}}$ and shares its weights with the context encoder. To keep the setup consistent with the JEPA objective, the VICReg branch also employs a deterministic predictor head (no dropouts) whose architecture and weights are tied to those of the main JEPA predictor.

Variance Loss ($\mathcal{L}_{var}$). The variance loss ensures that each embedding dimension maintains sufficient variance across the batch, preventing informational collapse:

$$\mathcal{L}_{var} = \frac{1}{d}\sum_{j=1}^{d} \max(0, \gamma - \sigma(\mathbf{u}_{:,j})) \tag{3}$$

where $\sigma(\mathbf{u}_{:,j})$ is the standard deviation of the j-th dimension across the batch, and γ is a constant threshold (typically $\gamma = 1$).

Covariance Loss ($\mathcal{L}_{cov}$). The covariance loss encourages decorrelation between different embedding dimensions, preventing redundant representations:

$$\mathcal{L}_{cov} = \frac{1}{d}\sum_{i\neq j} C_{ij}^2 \tag{4}$$

where $C \in \mathbb{R}^{d\times d}$ is the covariance matrix of the centered embeddings:

$$C = \frac{1}{B-1}(\mathbf{U} - \bar{\mathbf{U}})^\top(\mathbf{U} - \bar{\mathbf{U}}) \tag{5}$$

and $\bar{\mathbf{U}}$ denotes the batch mean. By penalizing the squared off-diagonal elements of the covariance matrix, this loss encourages each dimension to encode independent information.

3.3.4 TOTAL OBJECTIVE

The final optimization problem is defined as:

$$\min_{\theta,\phi} \mathcal{L}_{total} = \lambda_1\mathcal{L}_{llm} + \lambda_2\mathcal{L}_{jepa} + \lambda_3\mathcal{L}_{var} + \lambda_4\mathcal{L}_{cov} \tag{6}$$

where $\lambda_{1,2,3,4}$ are hyper-parameters that weigh the contribution of each objective.

3.4 COMPATIBILITY ACROSS MODEL ARCHITECTURES AND OBJECTIVES

The JEPA-DNA framework is model-agnostic and compatible with MLM and NTP objectives. Its primary requirement is an aggregation operator that compresses a sequence into a latent representation. While the $[CLS]$ token is "native" to the Transformer encoder architecture trained with MLM objectives (Devlin et al., 2019), JEPA-DNA also extends naturally to Transformer decoders, State Space Models (SSMs), and long-convolution backbones (e.g., HyenaDNA (Nguyen et al., 2023)) that are trained with NTP objectives. In these architectures, we use the embeddings of the last $[SEP]$ token of the sequence, and the same JEPA formulation is equally applicable when the aggregation operator is implemented as mean pooling over token embeddings, as in Nucleotide Transformer–style models (Boshar et al., 2025). By supervising the DNA sequence's global representation within the JEPA loss, regardless of the generative objective or specific encoder architecture, we enforce a global pooling mechanism, effectively compensating for the otherwise local or recurrent nature of these operators.

4 EXPERIMENTAL RESULTS

We evaluate the JEPA-DNA framework across diverse architectures and benchmarks to quantify its impact on representational quality.

4.1 EXPERIMENTAL SETUP

We provide the core methodological and implementation details below, while deferring extended specifications, training and implementation details to the Appendix (Section A.3).

Backbones & Predictor Architecture. To evaluate the compatibility of JEPA-DNA, we select three representative models to serve at its encoder backbone, spanning the dominant architectural paradigms in GFM research: (1) DNABERT-2 (Zhou et al., 2024), a 117M-parameter Transformer Encoder utilizing MLM and BPE tokenization; (2) NTv3 (Boshar et al., 2025), a 100M-parameter hybrid U-Net/Transformer model utilizing MLM over single nucleotides; and (3) HyenaDNA (Nguyen et al., 2023), a sub-quadratic operator model optimized via an NTP objective. These models vary in their sequence aggregation strategies, utilizing $[CLS]$ embeddings, mean pooling, and last tokens, respectively. The JEPA-DNA framework introduces a target encoder (EMA-updated version of the GFM backbone) and a lightweight predictor. For token-aggregated backbones (DNABERT-2, HyenaDNA), the predictor is a 4-layer Transformer that reconstructs the target's latent representations in a reduced embedding space. For mean-pooled backbones (NTv3), we employ a 3-layer MLP predictor that operates directly on the pooled sequence embedding. This dual-design ensures that the latent prediction objective is compatible with both local attention-based and global pooling-based architectures.

Baselines. In addition to the original pre-trained model (*original checkpoint*), we report two continually-trained baselines to isolate the original checkpoint selection, and separate the effect of continued optimization from that of the latent objective. Both continue training each backbone with the same native objective and dataset used for its own pre-training, and differ only in budget: the *step-matched* baseline is trained for the same number of iterations as JEPA-DNA, whereas the *compute-matched* baseline is trained for the same amount of compute time (see Appendix Table 4 for compute details).

Continual Pre-training Datasets and Evaluation Benchmarks. To ensure that performance improvements are attributable solely to the JEPA-DNA objective rather than exposure to novel data, we perform continual training using subsets of the original pre-training corpora for each backbone. For DNABERT-2, we utilize the multi-species corpus (human and five model organisms, 4.76M sequences); for NTv3, we use the Open Genome 2 corpus (Brixi et al., 2026) (random crops of 8,192 bp); and for HyenaDNA, we utilize the hg38 human reference (600k windows). By maintaining data parity with the original baselines, we isolate the impact of our latent-prediction framework as a general "representational uplift" mechanism. We utilize GFMBench-API (Larey et al., 2026) to evaluate representational quality across a standardized suite of genomic tasks. Our supervised evaluation includes nine tasks spanning regulatory syntax (GUE (Zhou et al., 2024)) and functional variants (VariantBenchmarks (Medvedev et al., 2025), Long Range Benchmark (LRB) (Kao et al., 2024)). Our zero-shot evaluation covers eight clinical and fitness-scoring benchmarks tasks from BEND (Marin et al., 2024), TraitGym (Benegas et al., 2025b), BRCA1 (Findlay et al., 2018), and ClinVar-based suites to measure emergent biological priors (Benegas et al., 2025a).

4.2 LINEAR EVALUATION

Since JEPA-DNA acts during continual pre-training, the quantity of interest is the representation it produces. To evaluate the linear separability of functional features, we perform linear probing by training a single-layer classifier on top of a frozen backbone.

The results in Table 1 demonstrate that JEPA-DNA is compatible with each of the evaluated backbone families and improves most tasks in all three (additional metrics are reported in Appendix Table 5). We observe consistent performance gains across all models, with JEPA-DNA improving performance in 6-8 out of 9 tasks for every backbone evaluated, regardless of the original pre-training paradigm (MLM or NTP) or sequence aggregation strategy (e.g., $[CLS]$, last token, or mean pooling).

Table 1: Linear probing performance (AUROC) across diverse GFM backbones, objectives, aggregation strategies, and self-supervised datasets (DS). Gain denotes the *relative* AUROC gain of JEPA-DNA over the original checkpoint / step-matched / compute-matched models. Bold values indicate the top performance within each backbone group. '†' symbol indicates overall state-of-the-art under linear probing.

Task	Original checkpoint	Step-matched	Compute-matched	JEPA-DNA (Ours)	Gain % AUROC
DNABERT-2 (Transformer + MLM + CLS; Multi-Species DS)					
GUE TF Binding	0.788	0.793	0.796	**0.839**	+6.47 / +5.81 / +5.42
GUE Promoter	0.884	0.927	0.927	**0.928**	+4.98 / +0.06 / +0.11
GUE Splice Site	0.653	0.670	0.672	**0.705**	+7.96 / +5.28 / +4.87
VB Coding Path.	**0.624**	0.622	0.605	0.610	-2.24 / -1.88 / +0.77
VB Exp. Effect	0.674	0.667	0.671	**0.689**†	+2.23 / +3.34 / +2.73
VB Common/Rare	**0.505**	0.500	0.502	0.504	-0.20 / +0.88 / +0.31
VB meQTL	0.623	0.607	0.606	**0.645**†	+3.53 / +6.25 / +6.45
VB sQTL	**0.585**	0.583	0.582	0.577	-1.37 / -1.01 / -0.93
LRB Causal eQTL	0.697	0.702	0.705	**0.711**†	+2.01 / +1.35 / +0.81
Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 DS)					
GUE TF Binding	0.837	0.838	0.838	**0.849**†	+1.43 / +1.33 / +1.33
GUE Promoter	0.907	0.907	0.907	**0.937**†	+3.31 / +3.30 / +3.30
GUE Splice Site	0.713	0.713	0.712	**0.733**†	+2.81 / +2.87 / +2.89
VB Coding Path.	0.668	0.667	0.667	**0.675**†	+1.05 / +1.17 / +1.16
VB Exp. Effect	0.642	0.642	0.642	**0.654**	+1.87 / +1.86 / +1.86
VB Common/Rare	0.502	0.502	0.502	**0.510**	+1.59 / +1.51 / +1.50
VB meQTL	0.576	0.576	0.576	**0.615**	+6.77 / +6.83 / +6.83
VB sQTL	0.591	**0.591**	0.591	0.590	-0.17 / -0.20 / -0.20
LRB Causal eQTL	0.690	0.690	0.690	**0.697**	+1.01 / +1.04 / +1.05
HyenaDNA (Long Convolutions + NTP + last-token; Reference-Genome DS)					
GUE TF Binding	0.698	**0.699**	0.696	0.638	-8.60 / -8.78 / -8.31
GUE Promoter	0.686	0.689	0.699	**0.763**	+11.22 / +10.77 / +9.18
GUE Splice Site	0.506	0.509	0.515	**0.558**	+10.28 / +9.53 / +8.37
VB Coding Path.	0.515	0.514	0.510	**0.559**	+8.54 / +8.82 / +9.54
VB Exp. Effect	0.580	**0.582**	0.581	0.547	-5.69 / -6.02 / -5.85
VB Common/Rare	0.503	0.502	0.503	**0.504**	+0.20 / +0.37 / +0.18
VB meQTL	0.503	0.505	0.505	**0.518**	+2.98 / +2.63 / +2.64
VB sQTL	0.550	0.552	**0.553**	0.552	+0.36 / +0.05 / -0.21
LRB Causal eQTL	0.565	0.562	0.563	**0.623**	+10.27 / +10.95 / +10.68

Notably, DNABERT-2, NTv3, and HyenaDNA exhibit AUROC gains of up to +7.96%, +6.77%, and +11.22%, respectively, alongside mean improvements across all tasks of +2.60%, +2.19%, and +3.29% compared to the original checkpoint, indicating that the latent-prediction objective reorganizes the embedding space to prioritize functional sequence properties across diverse genomic regimes.

The two continually-trained baselines further indicate that these gains are not a by-product of additional optimization. Continuing to train each backbone with its own native objective is not consistently beneficial: the step-matched and compute-matched baselines each improve only 15 of 27 task–backbone pairs, averaging $+0.17\%$ and $+0.22\%$ respectively, helping some tasks while degrading others. JEPA-DNA, in contrast, surpasses each of them on 22 of 27 comparisons, with mean relative gains of $+2.52\%$ and $+2.46\%$.

Across all evaluated backbones, JEPA-DNA achieves consistent performance gains in the majority of tasks (6-8 out of 9, depending on the backbone), with substantial improvements in regulatory and structural identification tasks, including promoter classification, splice site identification, meQTL and causal eQTL prediction. We hypothesize that these gains arise because such tasks depend primarily on a broader regulatory context, making them well-suited to the global, context-focused objective of JEPA-DNA rather than to the reliance on local sequence variation. In particular, this interpretation is consistent with prior work showing that meQTL prediction and causal eQTL tasks capture how genetic variants interact with global attributes such as transcription factor binding, histone modifications, and chromatin accessibility (Banovich et al., 2014; Wang et al., 2013) whereas tasks such as coding pathogenicity and sQTL prediction are often driven by more fine-grained, local

sequence changes (Zeng & Li, 2022). The stability of these improvements is further validated through a multi-seed analysis (see Table 6 in Appendix A.2).

We further observe that larger, higher-performing models tend to benefit more consistently from JEPA-DNA. For example, in NTv3, JEPA-DNA outperforms the baselines on 8 out of 9 supervised tasks, with less than 1% degradation on the remaining task. In contrast, for HyenaDNA, although JEPA-DNA yields substantial improvements (e.g., a 9%–11% gain on the GUE promoter task) and overall positive effects (improvements in 6 out of 9 tasks), it also introduces notable degradations in certain cases, such as an 8% decrease in performance on the GUE TF binding task.

Finally, we compare JEPA-DNA against seven leading GFMs evaluated in their original form. JEPA-DNA achieves state-of-the-art results under our linear probing settings on 7 out of 9 tasks. We indicate state-of-the-art performance with a '†' symbol in Table 1 and provide the full per-model breakdown and full comparison in Tables 7–8, respectively.

Table 2: Zero-shot performance of JEPA-DNA using the DNABERT-2 backbone (AUROC), compared against the original checkpoint and the two continually-trained native baselines. Gain denotes the *relative* AUROC gain of JEPA-DNA over the original checkpoint / step-matched / compute-matched models. Bold values denote superior zero-shot performance.

Task	Original checkpoint	Step-matched	Compute-matched	JEPA-DNA (Ours)	Gain % AUROC
BEND Expression Effect	0.543	0.530	0.536	**0.574**	+5.71 / +8.21 / +7.00
BEND Disease Variant	0.680	0.638	0.631	**0.762**	+12.06 / +19.48 / +20.68
TraitGym Complex	**0.517**	0.512	0.514	0.503	-2.71 / -1.80 / -2.06
TraitGym Mendelian	0.515	0.530	0.526	**0.540**	+4.85 / +1.93 / +2.57
Songlab ClinVar	**0.540**	0.518	0.516	0.527	-2.41 / +1.72 / +2.15
LOL-EVE Causal eQTL	0.515	0.522	0.523	**0.545**	+5.83 / +4.47 / +4.26
BRCA1	0.505	0.522	0.501	**0.527**	+4.36 / +0.93 / +5.12
LRB Pathogenic OMIM	0.537	0.516	0.503	**0.565**	+5.21 / +9.49 / +12.43

4.3 Zero-Shot Evaluation

We assess out-of-the-box semantic understanding by computing the cosine similarity between reference and variant sequence embeddings. We quantify performance via AUROC over the resulting similarity scores, allowing us to measure the model's capacity to rank functional variants without task-specific training.

Table 2 summarizes these results using DNABERT-2 as the representative backbone. Compared to the original checkpoint of DNABERT-2, JEPA-DNA improves zero-shot performance in 6 out of 8 tasks, with a mean AUROC gain of +4.1% and substantial improvements (ranging from 4% to 12%) across clinically significant benchmarks tasks, including BEND Disease Variant (+12.1%) and LRB Pathogenic OMIM (+5.21%).

While JEPA-DNA provides broad benefits, the task-specific gain vary between architectures (additional backbones and metrics are reported in Tables 9– 10). In zero shot tasks, where the model did not receive direct supervision on functional or biological meaning, all evaluated models struggle, and performance remains low. However, JEPA-DNA recovers additional signal and improves performance in some cases, for example, in BEND Expression Effect and TraitGym Mendelian tasks. Continual training alone yields little to no benefit, improving only 10–12 of 24 task-backbone pairs with average relative AUROC changes of -0.60% to -0.21% for step-matched and compute-matched setups, respectively. In contrast, JEPA-DNA outperforms these compute- and step-matched baselines on 17 of 24 pairs, achieving mean relative gains of $+3.65\%$ and $+3.31\%$, correspondingly.

4.4 Ablations and Additional Experiments

Ablations. We systematically decompose the JEPA-DNA framework to identify the primary drivers of performance gains, using DNABERT-2 as the backbone and evaluating it on three representative tasks. Table 3 details the incremental utility of our framework's components, where VICReg (VR) regularization emerges as essential for unlocking the benefits of the hybrid MLM+JEPA paradigm. Peak performance is achieved through the synergistic combination of regularized latent prediction and

Table 3: Factorial ablation of the JEPA-DNA components on DNABERT-2. Promoter, Splice and Express denote GUE Promoter, GUE Splice Site and BEND Expression Effect, respectively. Each entry reports *step-matched* (100K optimization steps) / *compute-matched* (equal GPU-hours) results; Avg. Gain is the mean *relative* AUROC gain over the baseline.

Configuration	**Components**					**AUROC** (step / compute)			**Avg. Gain**
	MLM	EMA	JEPA	VICReg	Sched. Mask	Promoter	Splice	Express.	% AUROC
Baseline (DNABERT-2)	×	×	×	×	×	0.884	0.653	0.543	—
MLM	✓	×	×	×	×	0.927/0.927	0.670/0.672	0.530/0.536	1.71/2.16
MLM+SM	✓	×	×	×	✓	0.927/0.927	0.669/0.669	0.530/0.530	1.66/1.64
MLM+EMA	✓	✓	×	×	×	0.929/0.929	0.674/0.674	0.532/0.532	2.09/2.09
MLM+SM+EMA	✓	✓	×	×	✓	0.927/0.927	0.669/0.669	0.530/0.530	1.67/1.64
MLM+VR+SM *(no JEPA)*	✓	✓	×	✓	✓	0.924/0.925	0.678/0.679	0.547/0.549	3.06/3.24
JEPA+VR+SM *(no MLM)*	×	✓	✓	✓	✓	0.919/0.920	0.680/0.676	0.562/0.565	3.86/3.85
MLM+JEPA	✓	✓	✓	×	×	0.927/0.927	0.669/0.670	0.529/0.529	1.58/1.63
MLM+JEPA+SM *(no VR)*	✓	✓	✓	×	✓	0.925/0.925	0.687/0.687	0.519/0.525	1.81/2.18
MLM+JEPA+VR *(no SM)*	✓	✓	✓	✓	×	0.928/0.928	0.689/0.689	0.565/0.565	4.85/4.85
JEPA-DNA (full)	✓	✓	✓	✓	✓	**0.928**	**0.705**	**0.574**	**6.22/6.22**

Scheduled Masking (SM). The grid is reported under both a step-matched and a compute-matched budget and includes a single-component elimination control for each part of the framework.

We also compare latent-space prediction against reconstruction-based paradigms for continual training. Table 11 in the Appendix shows that latent-space alignment (+3.86%) provides a more informative training signal than token reconstruction (+2.06%). Optimal results require a hybrid approach that leverages MLM tokens and local reconstruction alongside latent sequence alignment.

We further evaluate our architectural and optimization choices. Success in latent alignment depends on the predictor architecture. As shown in Table 12 in the Appendix, an attention-based predictor substantially outperforms the MLP variant, with performance peaking at a depth of four layers. We also adopt the target encoder, consistent with prior JEPA approaches, and employ it for inference. We also find that the SGDM algorithm is essential for training stability, as standard adaptive optimizers such as Adam or AdamW frequently lead to performance degradation (see Appendix Table 13). The extended ablation analysis is provided in Appendix A.2.

Additional Experiments. We briefly summarize five additional experiments. Compared against DNASimCLR (Yang et al., 2024) and three matched contrastive variants, JEPA-DNA is ahead on all evaluated tasks ($+7.57\%$ against $+4.06\%$ for the strongest variant, Table 14 in the Appendix). Additionally, Appendix A.2 details an embedding analysis (Figure 2), a full fine-tuning comparison (Table 15), a sequence-length stability analysis (Table 16) and a loss-weight sensitivity analysis (Table 17).

5 Conclusion

Summary. We introduced JEPA-DNA, a framework adapting the Joint Embedding Predictive Architecture to genomic sequences. By predicting latent representations in addition to reconstructing tokens, JEPA-DNA captures high-level biological context more effectively than generative objectives alone, consistently boosting backbone performance across supervised and zero-shot tasks.

Limitations and Future Work. While our work focused on representation quality, the behaviour of full fine-tuning remains partially addressed, as our end-to-end comparison covers only one backbone and a subset of tasks (Appendix Table 15). A broader study spanning full fine-tuning and parameter-efficient adaptation is deferred for future work. In addition, due to the substantial computational demands of large-scale genomic training, this study focused on the continual pre-training of existing state-of-the-art backbones. Future work will evaluate the JEPA objective for training models from scratch on massive, multi-species datasets to determine if it accelerates the emergence of biological priors. Furthermore, integrating recent advancements such as LeJEPA (Balestriero & LeCun, 2025), could potentially offer better scaling properties for the extremely long-range dependencies inherent in genomic data.

AI USE STATEMENT

In this work, we used generative AI tools to assist with validating the method implementation and to provide a secondary check of the authors' independently conducted results. We have not used generative AI tools to formulate the research questions, develop the research ideas or hypotheses, design the experimental setups, create the figures, structure the paper, or compile and format the bibliography. Additionally, we used generative AI tools to suggest and edit code and to correct and refine grammar when writing the paper. We have reviewed all AI-assisted work: the authors manually verified and tested all suggested code and reviewed all textual suggestions and analysis feedback. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study did not involve the recruitment of human participants or the collection of new human-subject, patient-level, or biological data. All experiments used previously released genomic resources and established research benchmarks. This work is intended solely for research and is not intended for clinical application. Neither the proposed method nor its predictions should be used for diagnosis, treatment, or other clinical decision-making. Genomic models may inherit biases and limitations from their training data, reference genomes, annotations, and evaluation benchmarks, and their performance may not generalize across populations or clinical settings. Any downstream clinical use would require independent validation and appropriate ethical, regulatory, privacy, and fairness assessments.

REPRODUCIBILITY STATEMENT

Appendix A.3 provides the implementation details required to reproduce our experiments, including model configurations, dataset preparation, training procedures, hyperparameters, masking schedules, and evaluation protocols. These details enable readers to train and evaluate the models using the same configurations reported in the paper and reproduce the presented results. In addition, the supplementary material includes a ZIP archive containing the code for continual pre-training of JEPA-DNA and for evaluating JEPA-DNA and the baseline models. The archive includes documentation and scripts for reproducing the reported experiments and results. Links to code on GitHub and checkpoint models on Hugging Face will be added to the paper upon acceptance.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Nicholas E Banovich, Xun Lan, Graham McVicker, Bryce Van de Geijn, Jacob F Degner, John D Blischak, Julien Roux, Jonathan K Pritchard, and Yoav Gilad. Methylation qtls are associated with coordinated changes in transcription factor binding, histone modifications, and gene expression levels. *PLoS genetics*, 10(9):e1004663, 2014.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful predictors of genome-wide variant effects. *Proceedings of the National Academy of Sciences*, 120(44): e2311219120, 2023.

Gonzalo Benegas, Carlos Albors, Alan J Aw, Chengzhong Ye, and Yun S Song. A dna language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, 43(12):1960–1965, 2025a.

Gonzalo Benegas, Gökcen Eraslan, and Yun S Song. Benchmarking dna sequence models for causal regulatory variant prediction in human genetics. *bioRxiv*, 2025b.

Gonzalo Benegas, Chengzhong Ye, Carlos Albors, Jianan Canal Li, and Yun S Song. Genomic language models: opportunities and challenges. *Trends in Genetics*, 41(4):286–302, 2025c.

Sam Boshar, Benjamin Evans, Ziqi Tang, Armand Picard, Yanis Adel, Franziska K Lorbeer, Chandana Rajesh, Tristan Karch, Shawn Sidbon, David Emms, et al. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. *BioRxiv*, pp. 2025–12, 2025.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics, Paris France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Garyk Brixi, Matthew G Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A Gonzalez, et al. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

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.

Deanna M Church, Valerie A Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M McLaren, Graham RS Ritchie, et al. Modernizing reference genome assemblies. *PLoS biology*, 9(7):e1001091, 2011.

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, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature methods*, 22(2):287–297, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

C. elegans Sequencing Consortium*. Genome sequence of the nematode c. elegans: a platform for investigating biology. *Science*, 282(5396):2012–2018, 1998.

Gregory M Findlay, Riza M Daza, Beth Martin, Melissa D Zhang, Anh P Leith, Molly Gasperini, Joseph D Janizek, Xingfan Huang, Lea M Starita, and Jay Shendure. Accurate classification of brca1 variants with saturation genome editing. *Nature*, 562(7726):217–222, 2018.

Arabidopsis Genome Initiative genomeanalysis@ tgr. org genomeanalysis@ gsf. de. Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *nature*, 408(6814):796–815, 2000.

Peter W Harrison, M Ridwan Amode, Olanrewaju Austine-Orimoloye, Andrey G Azov, Matthieu Barba, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, et al. Ensembl 2024. *Nucleic acids research*, 52(D1):D891–D899, 2024.

Roger A Hoskins, Joseph W Carlson, Kenneth H Wan, Soo Park, Ivonne Mendez, Samuel E Galle, Benjamin W Booth, Barret D Pfeiffer, Reed A George, Robert Svirskas, et al. The release 6 reference sequence of the drosophila melanogaster genome. *Genome research*, 25(3):445–458, 2015.

Kerstin Howe, Matthew D Clark, Carlos F Torroja, James Torrance, Camille Berthelot, Matthieu Muffato, John E Collins, Sean Humphray, Karen McLaren, Lucy Matthews, et al. The zebrafish reference genome sequence and its relationship to the human genome. *Nature*, 496(7446):498–503, 2013.

Hai Huang, Yann LeCun, and Randall Balestriero. Llm-jepa: Large language models meet joint embedding predictive architectures. In *International Conference on Learning Representations*, volume 2026, pp. 105717–105737, 2026.

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.

Chia Hsiang Kao, Evan Trop, McKinley Polen, Yair Schiff, Bernardo P de Almeida, Aaron Gokaslan, Thomas Pierrot, and Volodymyr Kuleshov. Advancing dna language models: The genomics long-range benchmark. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*, 2024.

W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at ucsc. *Genome research*, 12(6): 996–1006, 2002.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ariel Larey, Elay Dahan, Amit Bleiweiss, Raizy Kellerman, Guy Leib, Omri Nayshool, Dan Ofer, Tal Zinger, Dan Dominissini, Gideon Rechavi, et al. Gfmbench-api: A standardized interface for benchmarking genomic foundation models. *Biorxiv*, pp. 2026–02, 2026.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Elon Litman, Tyler Myers, Vinayak Agarwal, Ekansh Mittal, Orion Li, Ashwin Gopinath, and Timothy Kassis. Genejepa: A predictive world model of the transcriptome. *bioRxiv*, pp. 2025–10, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Frederikke Marin, Felix Teufel, Marc Horlacher, Dennis Madsen, Dennis Pultz, Ole Winther, and Wouter Boomsma. Bend: Benchmarking dna language models on biologically meaningful tasks. In *International Conference on Learning Representations*, volume 2024, pp. 15246–15281, 2024.

Aleksandr Medvedev, Karthik Viswanathan, Praveenkumar Kanithi, Kirill Vishniakov, Prateek Munjal, Clement Christophe, Tiago R Magalhães, Marco AF Pimentel, Ronnie Rajan, and Shadab Khan. Biotoken and biofm–biologically-informed tokenization enables accurate and efficient genomic foundation models. *bioRxiv*, pp. 2025–03, 2025.

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

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

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *Proceedings of machine learning research*, 235:43632, 2024.

Valerie A Schneider, Tina Graves-Lindsay, Kerstin Howe, Nathan Bouk, Hsiu-Chuan Chen, Paul A Kitts, Terence D Murphy, Kim D Pruitt, Françoise Thibaud-Nissen, Derek Albracht, et al. Evaluation of grch38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome research*, 27(5):849–864, 2017.

Courtney A Shearer, Rose Orenbuch, Felix Teufel, Christian J Steinmetz, Daniel Ritter, Erik Xie, Artem Gazizov, Aviv Spinner, Jonathan Frazer, Mafalda Dias, et al. A genomic language model for zero-shot prediction of promoter variant effects. *bioRxiv*, pp. 2024–11, 2024.

Dennis Wang, Augusto Rendon, and Lorenz Wernisch. Transcription factor and chromatin features predict genes associated with eqtls. *Nucleic acids research*, 41(3):1450–1463, 2013.

Minghao Yang, Zehua Wang, Zizhuo Yan, Wenxiang Wang, Qian Zhu, and Changlong Jin. Dnasimclr: a contrastive learning-based deep learning approach for gene sequence data classification. *BMC bioinformatics*, 25(1):328, 2024.

Tony Zeng and Yang I Li. Predicting rna splicing from dna sequence using pangolin. *Genome biology*, 23(1):103, 2022.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genomes. In *International Conference on Learning Representations*, volume 2024, pp. 41642–41665, 2024.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020.

A Appendix

This appendix provides supplementary information to support the findings and methodology presented in the main text.

Section A.1 quantifies the training and inference cost of the framework.

Section A.2 offers extended experimental results, including robustness (multi-seed) analysis, detailed performance breakdowns, and additional ablations, as well as comparisons against continually-trained baselines, a full fine-tuning study, loss-weight sensitivity, and an analysis of behaviour beyond the training context.

Section A.3 provides a comprehensive description of the experimental setup, covering backbone and architecture specifications, dataset and task descriptions, evaluation protocols, and specific implementation and training details required for reproducibility.

A.1 Computational Cost

Table 4 reports the training cost of JEPA-DNA relative to matched native continual training. The overhead stems from the additional forward passes through the EMA target encoder and the predictor, and ranges from $1.09\times$ to $7.88\times$ in time per step depending on the backbone. This overhead is confined entirely to the continual pre-training phase. At inference the predictor and the VICReg heads are discarded and only the EMA target encoder is retained, so runtime and memory are identical to those of the original backbone.

Table 4: Computational cost of JEPA-DNA relative to matched native continual training. The overhead is confined to pre-training; inference cost is unchanged.

Backbone	**Cost**	**Native MLM/NTP**	**JEPA-DNA**	**Ratio**
DNABERT-2	Time / step	0.955 s	2.661 s	2.78×
	Peak memory	21.34 GiB	49.42 GiB	2.32×
	GPU-hours (100K steps)	∼46.3 h	∼128.9 h	2.78×
NTv3-100M	Time / step	4.654 s	5.055 s	1.09×
	Peak memory	37.45 GiB	71.82 GiB	1.92×
	GPU-hours (32.5K steps)	∼40.2 h	∼43.6 h	1.09×
HyenaDNA	Time / step	0.631 s	4.966 s	7.88×
	Peak memory	3.80 GiB	15.11 GiB	3.98×
	GPU-hours (45K steps)	∼14.0 h	∼110.0 h	7.88×
All	**Inference time / memory**	—	—	**1.00**×

Table 5: Linear probing performance across diverse GFM backbones, objectives, aggregation strategies, and self-supervised datasets (DS). Bold values indicate the top performance within each backbone group. Gain denotes the *relative* AUROC gain over the corresponding original checkpoint baseline.

Task	Baseline			JEPA-DNA (Ours)			Gain
	MCC	AUROC	AUPRC	MCC	AUROC	AUPRC	% AUROC
DNABERT-2 (Transformer + MLM + CLS; Multi-Species DS)							
GUE TF Binding	0.429	0.788	0.773	**0.515**	**0.839**	**0.828**	+6.47%
GUE Promoter	0.607	0.884	0.877	**0.717**	**0.928**	**0.913**	+4.98%
GUE Splice Site	0.000	0.653	0.452	**0.139**	**0.705**	**0.500**	+7.96%
VB Coding Path.	0.002	**0.624**	**0.513**	**0.078**	0.610	0.509	-2.24%
VB Exp. Effect	0.254	0.674	0.647	**0.273**	**0.689**	**0.658**	+2.23%
VB Common/Rare	0.007	**0.505**	**0.504**	0.007	0.504	0.503	-0.20%
VB meQTL	0.032	0.623	0.552	**0.125**	**0.645**	**0.579**	+3.53%
VB sQTL	**0.121**	**0.585**	**0.587**	0.110	0.577	0.582	-1.37%
LRB Causal eQTL	**0.307**	0.697	0.717	0.296	**0.711**	**0.727**	+2.01%
Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 DS)							
GUE TF Binding	0.509	0.837	0.825	**0.530**	**0.849**	**0.843**	+1.43%
GUE Promoter	0.648	0.907	0.894	**0.754**	**0.937**	**0.923**	+3.31%
GUE Splice Site	-0.004	0.713	0.487	**0.163**	**0.733**	**0.506**	+2.81%
VB Coding Path.	0.017	0.668	0.541	**0.099**	**0.675**	**0.554**	+1.05%
VB Exp. Effect	0.223	0.642	0.612	**0.235**	**0.654**	**0.626**	+1.87%
VB Common/Rare	**0.003**	0.502	0.503	-0.001	**0.510**	**0.508**	+1.59%
VB meQTL	-0.008	0.576	0.496	**0.039**	**0.615**	**0.545**	+6.77%
VB sQTL	0.131	**0.591**	0.591	**0.135**	0.590	0.591	-0.17%
LRB Causal eQTL	0.269	0.690	0.675	**0.289**	**0.697**	**0.686**	+1.01%
HyenaDNA (Long Convolutions + NTP + last-token; Reference-Genome DS)							
GUE TF Binding	**0.278**	**0.698**	**0.684**	0.084	0.638	0.608	-8.60%
GUE Promoter	0.296	0.686	0.721	**0.434**	**0.763**	**0.728**	+11.22%
GUE Splice Site	-0.012	0.506	0.336	**0.000**	**0.558**	**0.375**	+10.28%
VB Coding Path.	-0.001	0.515	0.444	**0.019**	**0.559**	**0.486**	+8.54%
VB Exp. Effect	**0.110**	**0.580**	**0.563**	0.108	0.547	0.523	-5.69%
VB Common/Rare	0.010	0.503	0.500	**0.016**	**0.504**	**0.505**	+0.20%
VB meQTL	**0.021**	0.503	0.461	0.020	**0.518**	**0.471**	+2.98%
VB sQTL	0.085	0.550	**0.556**	**0.089**	**0.552**	0.556	+0.36%
LRB Causal eQTL	0.108	0.565	0.567	**0.189**	**0.623**	**0.615**	+10.27%

A.2 Extended Results

Complete Supervised Metrics. Table 5 reports the full set of metrics for the original checkpoint and JEPA-DNA, namely the Matthews Correlation Coefficient (MCC), AUROC and AUPRC, across all backbones and supervised tasks.

Robustness Analysis. We conduct a multi-seed analysis across all three backbones to evaluate the stability of the reported performance gains. Specifically, we repeat the linear-probe training over five random seeds for each backbone and report the mean and standard deviation. As shown in Table 6, JEPA-DNA yields consistent and stable improvements. For DNABERT-2, JEPA-DNA improves on 6 of 9 tasks, with gains of up to $+7.81\%$ on splice site prediction, and regresses on coding pathogenicity, common vs. rare and sQTL. For NTv3, JEPA-DNA improves on 8 of 9 tasks, the single exception being sQTL at -0.08%, which lies inside one standard deviation. For HyenaDNA, JEPA-DNA improves on 6 of 9 tasks and regresses on TF binding, expression effect and common vs. rare. We report these degradations explicitly as trade-offs of the smallest backbone on locally-driven motif tasks. Where gains occur they are large relative to seed noise. For example, the NTv3 promoter improvement of 0.029 absolute AUROC is roughly seventy times the corresponding standard deviation. Multi-seed results therefore preserve the trends observed in our single-run experiments.

Benchmarking Against Leading GFMs. To provide a comprehensive comparison against leading GFMs, Table 7 details the per-task linear probing performance of each individual baseline GFM, reported as reference points for frozen representation quality. Table 8 summarizes this comparison:

Table 6: Impact of JEPA-DNA on Representation Stability and Performance. We compare each baseline backbone against its JEPA-DNA enhanced counterpart using linear probing across five random seeds, for all three backbones. We report the mean AUROC and AUPRC with standard deviations. Bold values indicate the superior mean performance. Gain denotes the *relative* AUROC gain over the corresponding original checkpoint baseline.

Task	Baseline		JEPA-DNA (Ours)		Gain
	AUROC	AUPRC	AUROC	AUPRC	% AUROC
DNABERT-2 (Transformer + MLM + CLS; Multi-Species DS)					
GUE TF Binding	0.787 ± .001	0.773 ± .001	**0.839** ± .001	**0.828** ± .001	+6.61%
GUE Promoter	0.885 ± .001	0.877 ± .001	**0.928** ± .000	**0.913** ± .000	+4.86%
GUE Splice Site	0.653 ± .001	0.451 ± .002	**0.704** ± .002	**0.500** ± .003	+7.81%
VB Coding Path.	**0.622** ± .003	**0.511** ± .003	0.610 ± .005	0.511 ± .006	-1.93%
VB Exp. Effect	0.673 ± .002	0.646 ± .001	**0.688** ± .001	**0.656** ± .001	+2.23%
VB Common/Rare	**0.508** ± .004	**0.504** ± .006	0.505 ± .004	0.505 ± .003	-0.59%
VB meQTL	0.619 ± .005	0.551 ± .004	**0.645** ± .004	**0.581** ± .005	+4.20%
VB sQTL	**0.585** ± .001	**0.588** ± .001	0.579 ± .002	0.584 ± .002	-1.03%
LRB Causal eQTL	0.696 ± .001	0.717 ± .001	**0.708** ± .002	**0.726** ± .001	+1.72%
Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 DS)					
GUE TF Binding	0.837 ± .001	0.824 ± .001	**0.848** ± .001	**0.842** ± .001	+1.33%
GUE Promoter	0.907 ± .000	0.894 ± .000	**0.937** ± .000	**0.923** ± .000	+3.23%
GUE Splice Site	0.714 ± .002	0.488 ± .002	**0.733** ± .002	**0.506** ± .001	+2.71%
VB Coding Path.	0.665 ± .002	0.539 ± .002	**0.676** ± .002	**0.555** ± .003	+1.63%
VB Exp. Effect	0.642 ± .001	0.613 ± .001	**0.653** ± .004	**0.624** ± .005	+1.68%
VB Common/Rare	0.505 ± .003	0.505 ± .003	**0.513** ± .005	**0.511** ± .004	+1.76%
VB meQTL	0.578 ± .001	0.497 ± .001	**0.614** ± .006	**0.546** ± .008	+6.39%
VB sQTL	**0.591** ± .001	**0.591** ± .001	0.590 ± .002	0.592 ± .001	-0.08%
LRB Causal eQTL	0.689 ± .001	0.675 ± .000	**0.697** ± .001	**0.686** ± .001	+1.13%
HyenaDNA (Long Convolutions + NTP + last-token; Reference-Genome DS)					
GUE TF Binding	**0.694** ± .006	**0.679** ± .005	0.582 ± .041	0.567 ± .035	-16.07%
GUE Promoter	0.675 ± .014	0.710 ± .015	**0.761** ± .009	**0.730** ± .009	+12.82%
GUE Splice Site	0.515 ± .011	0.342 ± .008	**0.548** ± .011	**0.369** ± .008	+6.36%
VB Coding Path.	0.517 ± .003	0.445 ± .003	**0.548** ± .010	**0.472** ± .014	+6.02%
VB Exp. Effect	**0.577** ± .003	**0.561** ± .002	0.556 ± .007	0.535 ± .010	-3.63%
VB Common/Rare	**0.506** ± .006	**0.504** ± .005	0.499 ± .005	0.498 ± .006	-1.35%
VB meQTL	0.511 ± .008	0.460 ± .004	**0.513** ± .004	**0.462** ± .006	+0.35%
VB sQTL	0.550 ± .001	0.556 ± .001	**0.553** ± .002	**0.555** ± .002	+0.58%
LRB Causal eQTL	0.552 ± .011	0.558 ± .006	**0.628** ± .004	**0.619** ± .006	+13.64%

the *Best Baseline GFM* column reports the maximum score achieved by any of these models in their original form for a given task, and *Best JEPA-DNA* the peak performance reached by extending three of them with our framework, so the baseline column is an oracle over seven leading models against only three JEPA models, making this a stricter test of our approach. Under this linear probing protocol, JEPA-DNA establishes a new performance ceiling across the majority of tasks, spanning regulatory syntax to long-range causal effects, consistently outperforming the best-performing individual baselines and thereby elevating the representational limits of current genomic architectures.

Zero-Shot Performance for Additional Backbones. Tables 9 and 10 provide the full zero-shot results for DNABERT-2, Nucleotide Transformer v3, and HyenaDNA, extending the analysis presented in the main text. These results confirm that the performance gains observed in zero-shot tasks are consistent across disparate model architectures, pre-training objectives and datasets.

Extended Ablation Studies. Table 3 in the main text isolates the contribution of each component of the framework. For the compute-matched budget, native continual MLM, with neither an EMA target nor a latent objective, yields only $+2.16\%$, so roughly 65% of the full $+6.22\%$ gain originates from components beyond matched continual training. Removing the latent objective while retaining regularization and scheduled masking (MLM+VR+SM) reaches $+3.24\%$, so regularization clearly contributes, but adding latent prediction approximately doubles the gain. Conversely, the latent

Table 7: AUROC comparison of supervised tasks for genomic foundation models. Multiple backbone models are evaluated under a linear probing setup on diverse genomic prediction benchmarks to identify the current state of the art for each task under these settings.

Task	DNABERT	DNABERT-2	NTv3 (8M)	NTv3 (100M)	EVO 2 (1B)	HyenaDNA	Caduceus
GUE TF Binding	0.812	0.788	0.792	**0.837**	0.630	0.698	0.651
GUE Promoter	0.836	0.884	0.904	**0.907**	0.815	0.686	0.693
GUE Splice Site	0.574	0.653	0.640	**0.713**	0.657	0.506	0.418
VB Coding Pathogenicity	0.613	0.624	0.595	**0.668**	0.659	0.515	0.586
VB Expression Effect	0.666	**0.674**	0.617	0.642	0.628	0.580	0.632
VB Common vs. Rare	**0.589**	0.505	0.466	0.502	0.525	0.503	0.501
VB meQTL	**0.635**	0.623	0.541	0.576	0.560	0.503	0.453
VB sQTL	0.558	0.585	0.580	**0.591**	0.573	0.550	0.572
LRB Causal eQTL	0.694	**0.697**	0.672	0.690	0.662	0.565	0.665

Table 8: Linear probing AUROC comparison between the top-performing GFMs and their corresponding JEPA-DNA augmented counterparts. Bold values indicate the best result under this protocol.

Task	Len.	Best Baseline GFM	Best JEPA-DNA (Ours)
TF Binding	100	0.837	**0.849**
Promoter	300	0.907	**0.937**
Splice Site	400	0.713	**0.733**
Coding Pathogenicity	1024	0.668	**0.675**
Expression Effect	1024	0.674	**0.689**
Common vs. Rare	1024	**0.589**	0.510
meQTL	1024	0.635	**0.645**
sQTL	1024	**0.591**	0.590
Causal eQTL	8192	0.697	**0.711**

objective without any token-level loss (JEPA+VR+SM) already attains $+3.85\%$, exceeding every MLM-only variant, which indicates that the two objectives are complementary rather than redundant. Neither the EMA target nor scheduled masking accounts for the gain in isolation, as EMA leaves matched continual MLM essentially unchanged ($+2.16\%$ to $+2.09\%$) while scheduled masking pays only when coupled with the latent objective ($+4.85\%$ to $+6.22\%$). Finally, MLM+JEPA without VICReg falls below MLM alone ($+1.63\%$ against $+2.16\%$). Absent a variance term the predictor drives the context encoder toward a low-rank solution, so VICReg is not an independent booster but the enabling condition for latent prediction.

Table 11 compares training paradigms according to the granularity at which each objective is applied. All JEPA variants share the same global objective on the sequence representative and differ only in the loss applied at the token level. Extending the latent objective to the token level as well, so that the individual masked tokens are also predicted in latent space and no MLM loss is used, reaches $+3.86\%$, whereas retaining the native MLM loss at the token level attains $+6.22\%$. The native token-level objective therefore remains important: where the tokenization provides a discrete vocabulary, classifying the masked tokens is a stronger local signal than reconstructing their representations in latent space. The two objectives are best combined by keeping MLM at the token level and applying JEPA to the sequence representative, rather than by using latent prediction at both granularities.

Table 12 evaluates the predictor head. An attention-based predictor outperforms the MLP variant, and performance peaks at a depth of four layers.

All previous ablation tables (Tables 3, 11 and 12) use the SGDM optimizer, so optimizer choice is held fixed across them. In Table 13 we compare optimization algorithms and find that adaptive methods are harmful in this continual-training regime, with Adam and AdamW degrading performance relative to the baseline (-12.24% and -4.45%) and AdaBelief yielding only a marginal gain ($+1.22\%$), whereas SGDM realizes the full $+6.22\%$.

DNASimCLR Comparison. We compare our JEPA-DNA approach against DNASimCLR (Yang et al., 2024), a contrastive learning framework based on SimCLR (Chen et al., 2020). Due to the absence of publicly available code, we reimplement the method following the original description and refine the official SimCLR implementation accordingly, adapting it to the DNA domain. Specifically,

Table 9: Zero-shot variant-effect performance (via AUROC) across all backbones, compared against the original checkpoint and the two continually-trained native baselines. *Step-matched* and *Compute-matched* continue training each backbone with its own native objective and dataset for the same number of iterations, and for the same amount of compute time, as JEPA-DNA respectively. Gain denotes the *relative* AUROC gain of JEPA-DNA over the original checkpoint / step-matched / compute-matched models. Bold values indicate the top performance within each backbone group.

Task	Original checkpoint	Step-matched	Compute-matched	JEPA-DNA (Ours)	Gain % AUROC
DNABERT-2 (Transformer + MLM + CLS; Multi-Species DS)					
BEND Expression Effect	0.543	0.530	0.536	**0.574**	+5.71 / +8.21 / +7.00
BEND Disease Variant	0.680	0.638	0.631	**0.762**	+12.06 / +19.48 / +20.68
TraitGym Complex	**0.517**	0.512	0.514	0.503	-2.71 / -1.80 / -2.06
TraitGym Mendelian	0.515	0.530	0.526	**0.540**	+4.85 / +1.93 / +2.57
Songlab ClinVar	**0.540**	0.518	0.516	0.527	-2.41 / +1.72 / +2.15
LOL-EVE Causal eQTL	0.515	0.522	0.523	**0.545**	+5.83 / +4.47 / +4.26
BRCA1	0.505	0.522	0.501	**0.527**	+4.36 / +0.93 / +5.12
LRB Pathogenic OMIM	0.537	0.516	0.503	**0.565**	+5.21 / +9.49 / +12.43
Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 DS)					
BEND Expression Effect	0.514	0.513	0.512	**0.579**	+12.65 / +12.97 / +12.99
BEND Disease Variant	0.713	0.716	**0.716**	0.680	-4.63 / -4.96 / -4.99
TraitGym Complex	0.495	0.480	0.508	**0.521**	+5.25 / +8.46 / +2.61
TraitGym Mendelian	0.507	0.497	0.514	**0.542**	+6.90 / +8.97 / +5.50
Songlab ClinVar	0.473	0.479	0.472	**0.504**	+6.55 / +5.11 / +6.86
LOL-EVE Causal eQTL	0.537	0.536	0.536	**0.545**	+1.49 / +1.62 / +1.68
BRCA1	0.628	0.625	**0.636**	0.599	-4.62 / -4.09 / -5.87
LRB Pathogenic OMIM	0.505	0.511	0.497	**0.533**	+5.54 / +4.27 / +7.19
HyenaDNA (Long Convolutions + NTP + last-token; Reference-Genome DS)					
BEND Expression Effect	0.486	0.487	0.501	**0.535**	+10.08 / +9.91 / +6.87
BEND Disease Variant	0.504	0.507	**0.512**	0.502	-0.40 / -0.93 / -1.96
TraitGym Complex	0.503	0.492	0.501	**0.508**	+0.99 / +3.34 / +1.41
TraitGym Mendelian	0.496	0.518	0.513	**0.548**	+10.48 / +5.87 / +6.79
Songlab ClinVar	0.497	0.500	**0.513**	0.502	+1.01 / +0.42 / -2.22
LOL-EVE Causal eQTL	**0.559**	0.548	0.522	0.539	-3.58 / -1.65 / +3.23
BRCA1	0.511	0.496	**0.528**	0.483	-5.48 / -2.68 / -8.61
LRB Pathogenic OMIM	0.507	0.506	**0.511**	0.489	-3.55 / -3.43 / -4.24

DNA sequences are encoded as one-hot representations and processed using a ResNet50 backbone, treating sequences as image-like inputs. Training uses augmented pairs generated by randomly masking approximately 30% of each sequence, where positive pairs originate from the same sequence and negatives are drawn from other sequences within the batch. To ensure a fair comparison of training paradigms, we adopt a multi-species dataset and sequence lengths consistent with DNABERT-2. As shown in Table 14, JEPA-DNA with a DNABERT-2 backbone consistently outperforms DNASimCLR across all evaluated tasks, with particularly notable gains on splice site prediction, resulting in an average AUROC improvement of 7.57%. Because the published configuration differs from ours in backbone, initialization and training history, we further report three controlled variants. Increasing the contrastive vision backbone from ResNet-50 with 25.6M parameters to ResNet-101-2 with 126.9M parameters, beyond the 117M parameters of the DNABERT-2 backbone used by JEPA-DNA, does not increase performance (-1.53%). Replacing it with the same pretrained DNABERT-2 backbone used by JEPA-DNA, while retaining the contrastive loss and adapting the vision-style augmentations to token-level operations, recovers $+4.06\%$ but remains below JEPA-DNA ($+7.57\%$). Finally, adding the MLM objective to the contrastive one degrades it to -0.34%, whereas the same addition to the latent objective improves it (Table 3).

Embeddings Analysis. We defined a set of genomic loci on human chromosome 22 (GRCh38) using public Ensembl annotations (Harrison et al., 2024), including the gene annotation (GTF), regulatory feature track (GFF3), and motif feature track (GFF3) from the same release. From the GTF, we retained protein-coding and long non-coding RNA genes, representing each gene by the midpoint of its annotated span. Splice donor and acceptor sites were inferred from consecutive exons within the same transcript via intron boundaries. Regulatory and motif intervals were extracted from GFF3

Table 10: Zero-shot capabilities of JEPA-DNA augmented models against their original baselines. Bold values denote superior zero-shot performance within each backbone group. Gain denotes the *relative* AUROC gain over the corresponding original checkpoint baseline.

Task	**Baseline**		**JEPA-DNA (Ours)**		**Gain**
	AUROC	AUPRC	AUROC	AUPRC	% AUROC
DNABERT-2 (Transformer + MLM + CLS; Multi-Species DS)					
BEND Expression Effect	0.543	0.084	**0.574**	**0.091**	+5.71%
BEND Disease Variant	0.680	0.141	**0.762**	**0.262**	+12.06%
TraitGym Complex	**0.517**	**0.106**	0.503	0.100	-2.71%
TraitGym Mendelian	0.515	0.103	**0.540**	**0.110**	+4.85%
Songlab ClinVar	**0.540**	**0.566**	0.527	0.557	-2.41%
LOL-EVE Causal eQTL	0.515	0.083	**0.545**	**0.090**	+5.83%
BRCA1	0.505	0.210	**0.527**	**0.218**	+4.36%
LRB Pathogenic OMIM	0.537	0.002	**0.565**	0.002	+5.21%
Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 DS)					
BEND Expression Effect	0.514	0.079	**0.579**	**0.091**	+12.65%
BEND Disease Variant	**0.713**	**0.147**	0.680	0.131	-4.63%
TraitGym Complex	0.495	0.099	**0.521**	**0.104**	+5.25%
TraitGym Mendelian	0.507	0.106	**0.542**	**0.110**	+6.90%
Songlab ClinVar	0.473	0.523	**0.504**	**0.558**	+6.55%
LOL-EVE Causal eQTL	0.537	0.084	**0.545**	**0.087**	+1.49%
BRCA1	**0.628**	**0.359**	0.599	0.290	-4.62%
LRB Pathogenic OMIM	0.505	0.002	**0.533**	0.002	+5.54%
Long Convolutions + NTP + last-token; Reference-Genome DS					
BEND Expression Effect	0.486	0.072	**0.535**	**0.084**	+10.08%
BEND Disease Variant	**0.504**	**0.075**	0.502	0.074	-0.40%
TraitGym Complex	0.503	0.099	**0.508**	**0.102**	+0.99%
TraitGym Mendelian	0.496	0.103	**0.548**	**0.124**	+10.48%
Songlab ClinVar	0.497	0.542	**0.502**	**0.543**	+1.01%
LOL-EVE Causal eQTL	**0.559**	**0.089**	0.539	0.086	-3.58%
BRCA1	**0.511**	**0.216**	0.483	0.200	-5.48%
LRB Pathogenic OMIM	**0.507**	0.002	0.489	0.002	-3.55%

Table 11: Comparative analysis of training paradigms. All JEPA variants include the VICReg loss. When the token type is JEPA, the JEPA objective is applied to the representations of individual masked tokens, analogous to the I-JEPA formulation. Bold values indicate the optimal configuration. Gains are recomputed against the revised continual-training runs and are consistent with Table 3. Gain denotes the *relative* AUROC gain over the corresponding baseline.

Paradigm	**Token**	**Global**	**Sched.**	**Promoter**		**Splice Site**		**Expression**		**Avg. Gain**
	Type	**JEPA**	**Mask**	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	% AUROC
MLM Only										
MLM	MLM	×	×	0.929	0.914	0.674	0.467	0.532	0.084	+2.09%
MLM	MLM	×	✓	0.927	0.911	0.669	0.463	0.530	0.084	+1.67%
JEPA Only										
JEPA	JEPA	✓	✓	0.919	0.906	0.680	0.474	0.562	0.089	+3.86%
*Hybrid (JEPA-DNA, **Ours**)*										
JEPA	MLM	✓	×	0.928	0.908	0.689	0.482	0.565	0.088	+4.85%
JEPA	**MLM**	✓	✓	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+6.22%**

files, summarized by their midpoints, and labeled by feature type (e.g., promoter, enhancer, CTCF binding, open chromatin, transcription factor binding). This process yielded approximately 37,000 labeled windows across nine functional classes on chromosome 22. DNA sequence windows centered on these loci were extracted from the GRCh38 reference and passed through both the JEPA target encoder and context encoder for downstream embedding analysis. For visualization and quantitative evaluation, embeddings were projected to two dimensions using UMAP (Fig. 2), with a subsequent balanced per-class subsampling at 85% of the minimum class size (n = 702, nine biotypes, seed 42). A k-nearest-neighbor classifier applied in the original embedding space with stratified 5-fold

Table 12: Evaluation of predictor head's architecture and depth. Bold values denote the best-performing configuration. Gain denotes the *relative* AUROC gain over the corresponding baseline, not an absolute AUROC difference.

Predictor	**GUE Promoter**		**GUE Splice Site**		**BEND Expression**		**Avg. Gain**
Architecture	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	% AUROC
Baseline	0.884	0.877	0.653	0.452	0.543	0.084	—
MLP	0.900	0.889	0.648	0.450	0.564	0.088	+1.64%
2 layers	0.928	0.911	0.691	0.485	0.561	0.087	+4.70%
4 layers (Ours)	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+6.22%**
6 layers	0.928	0.910	0.655	0.451	0.555	0.086	+2.50%
8 layers	0.917	0.900	0.662	0.459	0.549	0.086	+2.07%

Table 13: Optimizer Sensitivity in JEPA-DNA Continual Pre-training. We evaluate the impact of different optimization algorithms on the stability and performance of JEPA-DNA using DNABERT-2 as the backbone. Bold values indicate peak performance. Gain denotes the *relative* AUROC gain over the corresponding baseline, not an absolute AUROC difference.

Optimizer	**Promoter**		**Splice Site**		**Expression**		**Avg. Gain**
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	% AUROC
Baseline	0.884	0.877	0.653	0.452	0.543	0.084	—
Adam Kingma & Ba (2014)	0.776	0.777	0.523	0.351	0.518	0.080	-12.24%
AdamW Loshchilov & Hutter (2017)	0.799	0.795	0.589	0.388	0.576	0.091	-4.45%
AdaBelief Zhuang et al. (2020)	0.901	0.894	0.662	0.454	0.545	0.086	+1.22%
SGDM Bottou (2010)	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+6.22%**

Table 14: DNASimCLR vs. JEPA-DNA across genomic tasks. We additionally report three controlled variants: scaling the contrastive vision backbone (ResNet-101-2), replacing it with the same pretrained DNABERT-2 backbone used by JEPA-DNA, and adding the MLM objective to the contrastive one. Gain denotes the *relative* AUROC gain over the published DNASimCLR configuration.

Learning Paradigm	**Promoter**		**Splice Site**		**Expression**		**Avg. Gain**
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	% AUROC
DNASimCLR Yang et al. (2024)	0.922	0.912	0.600	0.407	0.549	0.085	—
DNASimCLR (ResNet-101-2, 126.9M)	0.917	0.909	0.579	0.391	0.546	0.084	-1.53%
DNASimCLR w/ pretrained DNABERT-2 (117M)	0.921	0.919	0.680	0.500	0.543	0.071	+4.06%
DNASimCLR w/ DNABERT-2 + MLM	0.921	0.911	0.686	0.483	0.465	0.070	-0.34%
JEPA-DNA (Ours)	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+7.57%**

cross-validation achieved a weighted F1 score of 0.903, while the context-encoder under the same protocol reached 0.877, indicating slightly weaker class separability compared to the target-encoder.

Full Fine-Tuning. All results in this work follow the frozen linear-probing and zero-shot protocols, which isolate the quality of the learned representation. For completeness, Table 15 reports an end-to-end fine-tuning comparison on DNABERT-2. JEPA-DNA remains ahead on all four tasks, though on the saturated GUE benchmarks (0.90–0.98) the margin narrows to under 0.005 absolute AUROC, whereas on the non-saturated LRB causal eQTL task it reaches $+8.5\%$ relative.

Behaviour Beyond the Training Context. A known limitation of GFMs is degradation when evaluated on inputs longer than the context they were pre-trained on. In this sequence-length stability analysis, DNABERT-2 (self-supervised context of 512 tokens) and its JEPA-DNA counterpart are evaluated at increasing input lengths (beyond 2,500 bp) on BRCA1 and Enhancer–Target (Cheng et al., 2025) (Table 16). On Enhancer–Target, JEPA-DNA leads at every length ($+4.8\%$ to $+7.9\%$ relative). On BRCA1 the two are on par at 4,096 bp, after which the baseline loses 0.215 AUROC by 32,768 bp against 0.058 for JEPA-DNA, a $3.7\times$ slower degradation.

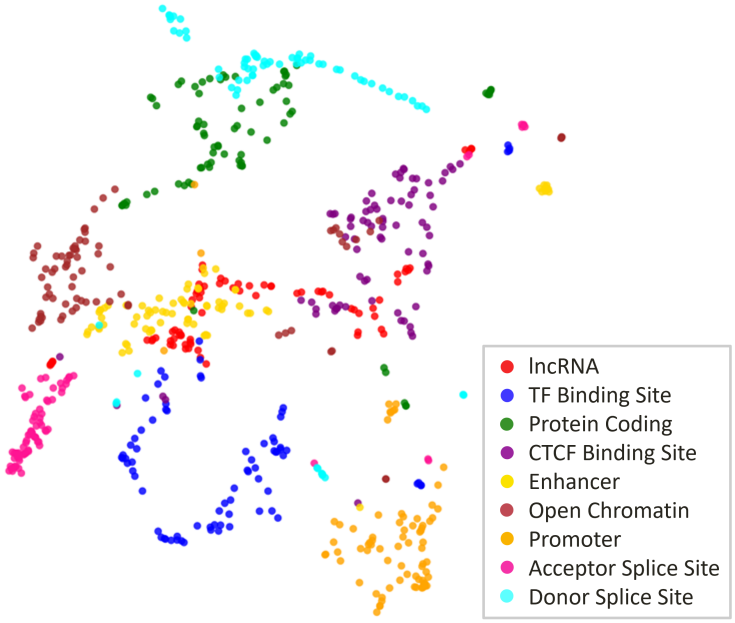

Figure 2: JEPA-DNA Embedding visualization colored by Ensembl annotations.

Table 15: Full fine-tuning of DNABERT-2 with and without JEPA-DNA continual training (AUROC), reporting both the absolute difference and the relative gain. The GUE tasks are saturated (0.90–0.98), where end-to-end fine-tuning largely washes out differences between the frozen encoders; on the non-saturated LRB causal eQTL task the advantage of the JEPA-DNA representation is retained.

Task	Baseline	JEPA-DNA	Rel. Gain
GUE TF Binding	0.9025	**0.9073**	+0.53%
GUE Promoter (all)	0.9818	**0.9843**	+0.25%
GUE Splice Site	0.9738	**0.9744**	+0.07%
LRB Causal eQTL	0.6669	**0.7237**	+8.51%

Table 16: Sequence-length sweep for DNABERT-2 (self-supervised training context of 512 tokens) evaluated at input lengths well beyond its pre-training window (AUROC). We present this as a stability and failure-mode analysis rather than as evidence of long-range reasoning.

Task	Model	4,096 bp	8,192 bp	16,384 bp	32,768 bp
BRCA1	Baseline	0.5485	0.4468	0.4864	0.3333
	JEPA-DNA	0.5470	**0.5280**	**0.4881**	**0.4892**
Enhancer–Target	Baseline	0.4467	0.4817	0.6077	0.6071
	JEPA-DNA	**0.4683**	**0.5179**	**0.6546**	**0.6548**

Loss-Weight Sensitivity. Table 17 reports the sensitivity of JEPA-DNA to the loss coefficients, perturbing each by $\pm 50\%$ around its default while holding the others fixed, on GUE Splice Site. The tuned configuration remains the strongest, but the method is not brittle: every perturbed setting still exceeds both the original backbone (0.653) and the continual MLM-only baseline (0.670), and the spread across all eight perturbations is 0.021 AUROC.

A.3 Extended Implementation Details

GFM Backbones Details. We demonstrate the JEPA-DNA framework across multiple genomic backbone models:

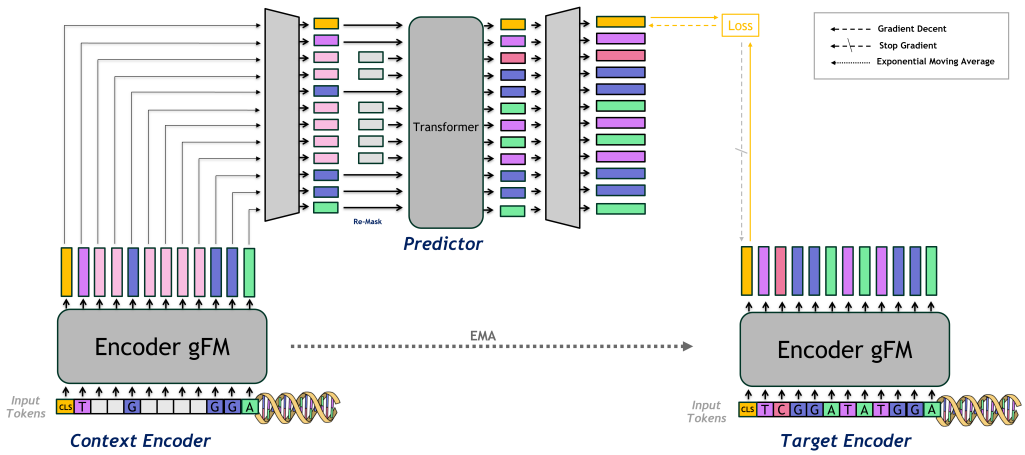

Figure 3: The JEPA-DNA architecture using a CLS-based representation, as implemented with DNABERT-2.

- **DNABERT-2 (Zhou et al., 2024):** A bidirectional encoder based on the BERT architecture (12 layers, 768 hidden dim, 12 heads, ∼117M parameters). It utilizes Byte Pair Encoding (BPE) and supports sequences up to 512 tokens. Sequence representations are derived from the `[CLS]` token.
- **NTv3 (Boshar et al., 2025):** A ∼100M-parameter hybrid model featuring a convolutional U-Net tower and a 6-layer Transformer stack. It is trained via MLM and utilizes a single nucleotide tokenization. Sequence-level representations are obtained via mean pooling over all token embeddings.
- **HyenaDNA (Nguyen et al., 2023):** An autoregressive model (∼600K parameters) utilizing long-convolutional blocks (Hyena operators) instead of attention. It is pre-trained via Next-Token Prediction (NTP) and derives sequence representations from the last `[SEP]` token embedding.

JEPA-DNA Predictor Configuration. The JEPA predictor is designed to be lightweight relative to the backbone to prevent a capacity bypass. For Transformer-based backbones, a lightweight 4-layer Transformer encoder operating in a reduced latent space whose dimensionality is set to one half of the backbone hidden size, with 3 attention heads. The predictor employs a pre-norm architecture with GELU activations and frozen sinusoidal positional embeddings. Input embeddings from the context encoder are first projected to the predictor dimension, with masked target positions replaced by learnable mask tokens before adding positional embeddings following the re-masking strategy described in Section 3.2. The predictor output for the sequence representative is then projected back to the original backbone hidden dimension for computing the prediction loss against the target encoder representation. For backbones where the sequence representation is obtained via mean pooling, we instead apply a 3-layer MLP predictor directly to the pooled sequence embedding, so that the sequence representative itself is processed by the predictor rather than being implicitly aggregated through attention.

Fig. 3 illustrates an instantiation of the JEPA-DNA architecture when using a CLS-based representation.

Table 17: Sensitivity to the loss weights on GUE Splice Site (AUROC). Each coefficient is perturbed by $\pm 50\%$ around its default, one at a time. *Original* denotes the original DNABERT-2 checkpoint, *Step* and *Compute* denote the step-matched and compute-matched continual MLM baselines, and *JEPA-DNA* the reported model with no perturbation of the loss weights.

	Baselines			Loss-weight perturbations								
	Original	Step	Compute	$\lambda_{\text{cov}} \times 0.5$	$\lambda_{\text{cov}} \times 1.5$	$\lambda_{\text{var}} \times 0.5$	$\lambda_{\text{var}} \times 1.5$	$\lambda_{\text{JEPA}} \times 0.5$	$\lambda_{\text{JEPA}} \times 1.5$	$\lambda_{\text{LLM}} \times 0.5$	$\lambda_{\text{LLM}} \times 1.5$	JEPA-DNA
AUROC	0.653	0.670	0.672	0.681	0.679	0.685	0.675	0.679	0.680	0.677	0.696	0.705

Continual Training Datasets. The JEPA-DNA framework is evaluated under a strict "data-parity" regime, where the continual training phase utilizes the exact same genomic sources as the original backbones' pre-training. This ensures that the framework is evaluated on its ability to extract more structured features from existing data rather than benefiting from additional genomic information. We construct three continual-training datasets for JEPA-DNA, each aligned with the pre-training corpus of its corresponding backbone. For DNABERT-2, we follow the original multi-species setting and use training samples built from the human reference genome (GRCh38/hg38) (Schneider et al., 2017) together with genomic data from five model organisms: mouse (Church et al., 2011), zebrafish (Howe et al., 2013), fruit fly (Hoskins et al., 2015), nematode (elegans Sequencing Consortium*, 1998), and thale cress (genomeanalysis@ tgr. org genomeanalysis@ gsf. de, 2000). All genome sequences are obtained from UCSC genome Browser (Kent et al., 2002), restricted to valid nucleotides, and chunked into fixed windows of 3200 bp with 50% overlap, yielding approximately 4.76×10^6 sequences. For NTv3, we use the Open Genome 2 corpus (Brixi et al., 2026), consisting of approximately 384×10^6 training records. Sequences are filtered to valid nucleotides, and each training sample is formed by taking a random crop of up to 8,192 nucleotides from a record. For HyenaDNA, we derive approximately 6×10^5 training windows from the full hg38 human reference FASTA (Schneider et al., 2017), each of length 8,192 bp with 50% overlap between consecutive windows, again restricting to valid nucleotides.

Evaluation Benchmarks. For downstream evaluation, we test on a diverse suite of supervised and zero-shot tasks derived from established genomic benchmarks (Larey et al., 2026). Our supervised evaluation includes three standard classification tasks from the GUE benchmark (Zhou et al., 2024): promoter prediction, transcription factor binding site prediction, and splice site prediction. Additionally, we utilize variant effect prediction tasks from VariantBenchmarks (Medvedev et al., 2025), covering coding pathogenicity classification, common vs. rare variant identification, and quantitative trait loci (QTL) prediction for expression, methylation (meQTL), and splicing (sQTL) effects. We also evaluate on the causal eQTL task from the Long Range Benchmark (LRB) (Kao et al., 2024), which involves high-context inputs. For zero-shot evaluation, we assess performance on variant effect prediction without task-specific fine-tuning. This includes the BEND benchmark (Marin et al., 2024) for expression and disease-associated variants, and TraitGym (Benegas et al., 2025b) for predicting Mendelian and complex traits. Furthermore, we evaluate clinical pathogenicity prediction using the Song-Lab ClinVar dataset (Benegas et al., 2025a) and the non-coding pathogenic OMIM task from LRB (Kao et al., 2024). In addition, we include variant-effect prediction on BRCA1 saturation mutagenesis (Findlay et al., 2018), and a LOL-EVE causal eQTL task (Shearer et al., 2024), providing further coverage of clinically and functionally relevant variant effect prediction tasks.

Evaluation Protocol. We employ two primary evaluation strategies to assess the quality of learned representations:

- Linear Probing: To isolate the quality of the pre-trained features, we keep the backbone frozen and train only a linear classifier on the extracted sequence representations. Performance is measured using three complementary metrics: (1) Area Under the Receiver Operating characteristic curve (AUROC), which measures the model's ability to discriminate between classes across all classification thresholds; (2) Area Under the Precision-Recall Curve (AUPRC), which is particularly informative for imbalanced datasets common in genomics tasks; and (3) Matthews Correlation Coefficient (MCC), a balanced metric that accounts for all four confusion matrix categories and remains robust even when class distributions are highly skewed.
- Zero-Shot Inference: We evaluate the model's out-of-the-box semantic understanding by computing the cosine similarity between embeddings of reference and variant representative sequences. For variant effect prediction tasks (e.g., ClinVar or TraitGym tasks), we extract sequence representations for both the reference and mutant sequences and measure their embedding angular distance. We assess the model's ability to rank functional variants without any task-specific training by computing AUROC and AUPRC over the similarity scores.

Training & Implementation Details. All models are implemented in PyTorch and trained within a unified JEPA-DNA framework. DNABERT-2 and NTv3 are trained on a single NVIDIA B300

GPU, using an Intel Xeon 6776P CPU with 2.0 TiB system RAM, and each model was trained for approximately five days. HyenaDNA is trained on a single NVIDIA A10 GPU, using an AMD EPYC 7R32 CPU with 200 GiB system RAM, and was trained for approximately ten days.

The context and target encoders are both initialized from pre-trained weights, while the predictor network is initialized from scratch using a truncated normal distribution with standard deviation 0.02 for all linear layer weights and zero initialization for biases.

For the masking strategy, we use a span-based scheduler that evolves over the course of training. At the beginning of training, we sample 1–2 contiguous target regions per sequence, covering 10–30% of the sequence length. Both the number of spans and the overall masking ratio are then increased linearly (with rounding) so that, by the end of training, we sample 1–4 contiguous target regions covering 30–50% of the sequence. The context encoder processes the masked sequence while the target encoder (updated via exponential moving average) processes the full unmasked sequence.

JEPA-DNA Continual training follows a two-phase schedule optimized for continual training and shared across all backbones.

- **Phase 1 (Predictor warmup).** For the first 1,000 steps, the encoder is frozen and only the predictor is trained, using a learning rate of 1×10^{-5} for DNABERT-2 and HyenaDNA and approximately 1.06×10^{-5} for NTv3. This warmup allows the predictor to learn meaningful initial representations before jointly updating the encoder.
- **Phase 2 (Full training).** The encoder is then unfrozen and trained jointly with the predictor using a 500-step linear warmup, followed by decay. For DNABERT-2 and HyenaDNA, the learning rate is linearly increased from 3×10^{-6} to a peak of 5×10^{-6} and then decayed with a cosine schedule to 1×10^{-6}. For NTv3, we adopt a lower learning-rate regime with phase-2 learning rates of 10^{-6} to 10^{-8}, reflecting its higher training sensitivity.

We use stochastic gradient descent (SGD) (Bottou, 2010) with momentum 0.9 and weight decay 0.01 for all models, with a batch size of 32 and gradient accumulation over 4 iterations (effective batch size per step is 128). Each backbone is trained for up to 100,000 optimization steps (each step comprising 4 gradient accumulation iterations) with an early stopping mechanism. At every checkpoint after a warmup we compute d_{ema}, the mean L_2 distance between the context-encoder and EMA-target sequence embeddings on a fixed set of clean, unmasked validation sequences. Training is stopped only when (i) the previous checkpoint is a local minimum, that is, its d_{ema} is at most $1.15\times$ the minimum over the previous four checkpoints, and (ii) the current d_{ema} is at least $3\times$ that minimum, in which case we restore the local-minimum checkpoint. While training is stable the online encoder and its EMA target remain closely aligned, so d_{ema} stays low and a sharp rise after such a quiet period marks a genuine context–target divergence. Under this rule DNABERT-2 never triggers early stopping and runs to the full budget, whereas NTv3 and HyenaDNA stop early. The target encoder is updated via an exponential moving average (EMA) with momentum scheduled from 0.996 to 1.0 over the course of training.

The training objective combines multiple losses:

- Latent Predictive Loss (JEPA Loss): Cosine similarity loss between the predicted and target sequence representations, used with weight 1.0, encouraging the context encoder to capture global sequence semantics from partial observations.
- LLM Loss: For bidirectional backbones, we apply a standard MLM objective on the masked token positions, reusing the JEPA encoder forward and masking. For autoregressive backbones, we instead use a next-token prediction (NTP) loss with an additional encoder forward under causal masking. The LLM loss term (MLM or NTP) is included with a weight of 0.1 in the total loss.
- Variance Loss: Hinge-based variance loss with weight 25.0 to prevent representation collapse by ensuring variance above a threshold (1.0) across the batch dimension. The variance is computed on the sequence representation embeddings from both the context encoder output and the predictor output. To avoid artificial variance that does not reflect true representational diversity, we employ two strategies: (1) variance calculations are performed via an additional forward pass through the context encoder and predictor in evaluation mode, eliminating stochastic effects from dropout and random masking; and (2) all sequences within each batch

are truncated to a uniform length, preventing spurious variance arising from heterogeneous padding patterns.

- Covariance Loss: With weight 0.5 to decorrelate embedding dimensions and encourage diverse feature learning.

For supervised downstream evaluation, we utilize a linear probing approach where the encoder weights are frozen and a linear classification head is trained on the sequence representations. For standard single-sequence tasks, the projection layer operates directly on the sequence embedding. In variant effect prediction, the reference and variant sequences are processed independently, and their corresponding sequence embeddings are concatenated prior to classification. To ensure rigorous comparison, all supervised tasks follow a unified training protocol: the classifier is trained for 3 epochs using the AdamW optimizer with a learning rate of 3×10^{-5}, a weight decay of 0.01, and a batch size of 32. For both supervised and zero-shot tasks, when sequence length exceeds the context window of the underlying backbone, inputs are truncated to retain the central region, using 2,500 bp (approximately 512 BPE tokens) for DNABERT-2 and 8,192 bp for NTv3 and HyenaDNA.