JEPA-DNA: Grounding Genomic Foundation Models through Joint-Embedding Predictive Architectures

Ariel Larey[1] **Elay Dahan**[2] **Amit Bleiweiss**[3] **Raizy Kellerman**[4]
Guy Leib[4] **Omri Nayshool**[4] **Dan Ofer**[4] **Tal Zinger**[4] **Dan Dominissini**[4]
Gideon Rechavi[4] **Nicole Bussola**[5] **Simon Lee**[5] **Shane O'Connell**[5]
Dung Hoang[5] **Marissa Wirth**[5] **Alexander W. Charney**[5]
Nati Daniel[1,*] **Yoli Shavit**[1,*]

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. Our framework establishes a new state-of-the-art for GFMs, surpassing the best existing models by bridging generative precision with latent semantic grounding. Through extensive ablation studies, we further characterize the synergistic interplay between generative and latent objectives. Our code is publicly available at https://github.com/NVIDIA-Digital-Bio/JEPA-DNA.

1 Introduction

Genomic Foundation Models (GFMs) have significantly advanced our ability to model the complex interactions within DNA sequences [1]. Architectures such as DNABERT-2 [2], Nucleotide Transformer (NT) [3], HyenaDNA [4], and Evo [5] adapt Large Language Models (LLMs) to genomic data, utilizing context windows from several kilobases to megabase scales. 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 [2, 3], generative objectives focus heavily on token-level reconstruction. This localized supervision can lead models to over-allocate

*Co-corresponding authors: ndaniel@nvidia.com, yolis@nvidia.com. [1]Applied AI Architecture, NVIDIA, Israel. [2]Worldwide Field Ops, NVIDIA, Israel. [3]Developer Programs, NVIDIA, Israel. [4]Cancer Research Center and Wohl Institute of Translational Medicine, Sheba Medical Center, Tel Hashomer, Israel. [5]Windreich Department of AI and Human Health, Icahn School of Medicine at Mount Sinai, New York, USA.

Preprint.

capacity to high-frequency sequence "noise" (e.g. repetitive elements or neutral polymorphisms) at the expense of global functional context. Consequently, GFMs may achieve high syntactic fidelity without internalizing the high-level regulatory logic or 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) [6] into genomic training. Unlike generative objectives that operate in the raw token space, the JEPA paradigm predicts the latent representations of masked segments. JEPA-DNA is the first to adapt Joint-Embedding Predictive Architectures to high-resolution, multiscale genomic sequences. By coupling token-level recovery with latent-space prediction, JEPA-DNA encourages the learning of abstract, functional features invariant to low-level sequence noise.

Our approach is uniquely versatile, serving as a continual pre-training phase to "ground" existing GFMs across disparate architectures and objectives. This latent grounding acts as a corrective layer, anchoring a model's token-level knowledge to a semantic world model of genomic function. We evaluate JEPA-DNA across 17 genomic benchmark tasks, utilizing linear probing and zero-shot protocols to isolate representational quality [7]. Our empirical results demonstrate that latent grounding consistently elevates performance, establishing a new state-of-the-art for GFMs across supervised linear probing tasks. Furthermore, we characterize the technical requirements for this paradigm through extensive ablations on the training objectives, predictor architectures, and optimization strategies required to maintain stability.

In summary, our contributions are as follows:

- We introduce a novel continual training framework for genomic foundation models that shifts the learning objective from literal token reconstruction to latent feature alignment via a Joint-Embedding Predictive Architecture.
- 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 [2], NTv3 [8], and HyenaDNA [4]), tokenization methods, and generative objectives (MLM and NTP).
- We establish a new state-of-the-art across supervised linear probing tasks, empirically proving 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 (GFMs)

The development of GFMs has been largely inspired by the success of Large Language Models (LLMs) in Natural Language Processing. Early iterations, such as DNABERT [9], adapted the BERT architecture [10] replacing its original WordPiece subword tokenizer with k-mer tokenization to capture bidirectional context within the genome. More recently, DNABERT-2 [2] and the Nucleotide Transformer [3] 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 [4] utilizes the Hyena operator to process sequences at single-nucleotide resolution across long contexts. Similarly, Evo2 [11] leverages a StripedHyena 2 backbone to push the context size to megabase scales. In parallel, Nucleotide Transformer v3 (NTv3) [8] 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 [12] because the loss function is applied solely in the potentially "noisy" token space.

Alternative self-supervised strategies have sought to move beyond literal token reconstruction by incorporating evolutionary or contrastive constraints. Models such as GPN-MSA [13] 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 [14] 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. By utilizing a predictive latent objective, our approach bypasses the need for negative samples or complex alignment pipelines, offering a more scalable path toward grounding genomic foundation models in biological semantics.

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 representation of a masked segment rather than its literal pixels [15]. 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 [16], 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 [17]. However, GeneJEPA operates on gene-expression vectors, designed to learn and reason about gene-gene relationships within a cell. In contrast, JEPA-DNA focuses on the primary genomic sequence: the "blueprint" itself. 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. By applying the JEPA paradigm directly to raw sequences, we enable the learning of "world models" [6] for genomic function.

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 $\hat{\mathbf{z}}_{\text{glob}} = \mathcal{A}(\hat{\mathbf{z}})$.

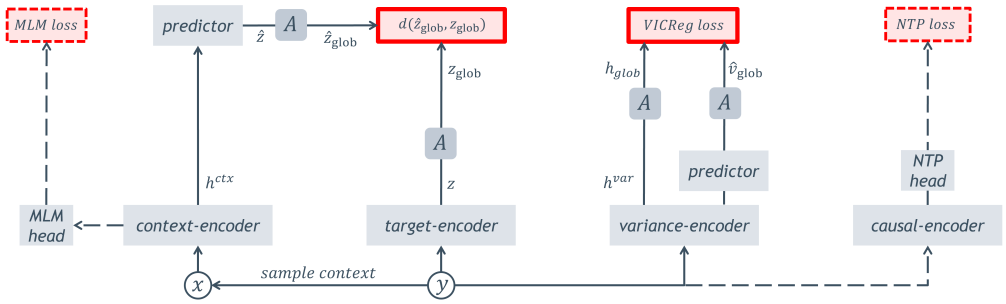

Figure 1: The JEPA-DNA Architecture.

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 [15]. 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:

$$\mathcal{L}_{llm} = -\sum_{i \in \mathcal{M}} \log P(t_i \mid \mathbf{h}_i^{ctx}) \tag{1}$$

where $P(t_i)$ denotes the predicted probability of the genomic token t_i at position i given the contextual hidden state $\mathbf{h}_i^{ctx}$. 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 [18]. 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 $\bar{\mathbf{Z}}$ 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 [10], JEPA-DNA also extends naturally to Transformer decoders, State Space Models (SSMs), and long-convolution backbones (e.g., HyenaDNA [4]) 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 [8]. 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. We compare our approach against leading GFMs baselines to demonstrate the universal representational gain provided by our latent-prediction objective.

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.2)

Backbones & Predictor Architecture. To evaluate the universality 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 [2], a 117M-parameter Transformer Encoder utilizing MLM and BPE tokenization; (2) NTv3 [8], a 100M-parameter hybrid U-Net/Transformer model utilizing MLM over single nucleotides; and (3) HyenaDNA [4], 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.

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 [11] (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 universal "representational uplift" mechanism.

We utilize GFMBench-API [7] to evaluate representational quality across a standardized suite of genomic tasks. Our supervised evaluation includes 9 tasks spanning regulatory syntax (GUE [2]) and functional variants (VariantBenchmarks [19], Long Range Benchmark (LRB) [20]). Our zero-shot evaluation covers 8 clinical and fitness-scoring benchmarks tasks from BEND [21], TraitGym [22], BRCA1 [23], and ClinVar-based suites to measure emergent biological priors [24].

4.2 Linear Evaluation

To evaluate the linear separability of functional features, we perform linear probing by training a single-layer classifier on top of a frozen backbone.

4.2.1 Architectural Universality

Our evaluation spans disparate backbone architectures, training objectives, and aggregation strategies, using three complementary metrics: Area Under the Receiver Operating Characteristic curve (AUROC), Area Under the Precision-Recall Curve (AUPRC), and the Matthews Correlation Coefficient (MCC).

Table 1: Linear probing performance across diverse GFM backbones, objectives, aggregation strategies, and self-supervised datasets (DS). Bold values indicating the top performance within each backbone group.

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	-0.19%
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%

The detailed results in Table 1 demonstrate that JEPA-DNA serves as an architecture-agnostic enhancement. 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). Notably, DNABERT-2, NTv3, and HyenaDNA exhibit AUROC gains of up to +8.0%, +6.8%, and +11.2%, respectively, alongside mean improvements across all tasks of +2.6%, +2.19%, and +3.28%, indicating that the latent-prediction objective reorganizes the embedding space to prioritize functional sequence properties across diverse genomic regimes.

Across all evaluated backbones, JEPA-DNA consistently improves performance on promoter recognition, splice-site classification, meQTL prediction, and causal eQTL tasks. 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 [25, 26] whereas tasks such as coding pathogenicity and sQTL prediction are often driven by more fine-grained, local sequence changes [27].

We further observe that larger, higher-performing models tend to benefit more consistently from JEPA-DNA. For example, in NTv3, JEPA-DNA outperforms the baseline 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., an 11% gain on the GUE promoter task) and overall positive effects (improvements in 7 out of 9 tasks), it also introduces notable degradations in certain cases, such as an 8% decrease in performance on GUE TF binding task.

The stability of these improvements is further validated through a multi-seed analysis (see Appendix A.1). As shown in Table 6, JEPA-DNA achieves consistent performance gains across the majority of tasks (6 out of 9), with substantial and stable improvements in regulatory and structural identification tasks, including TF binding, promoter classification, splice site identification, and meQTL prediction.

4.2.2 Benchmarking Against the State-of-the-Art

Table 2: Linear probing comparison between the top-performing state-of-the-art GFMs and their corresponding JEPA-DNA augmented counterparts. Bold values indicate the overall state-of-the-art.

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**

We further compare JEPA-DNA against seven leading genomic foundation models (GFMs). In Table 2, the *Best Baseline GFM* represents the maximum score achieved by any of these models in their original form for a given task (full individual metrics are available in Appendix Table 8). We then report the peak performance reached by extending three of these GFMs with the JEPA-DNA framework. JEPA-DNA establishes a new performance ceiling across the majority of tasks, spanning regulatory syntax to long-range causal effects. By integrating our framework with existing GFMs, we consistently outperform the best-performing individual baselines in most tasks, demonstrating that JEPA-DNA elevates the representational limits of current genomic architectures.

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 and AUPRC over the resulting similarity scores, allowing us to measure the model's capacity to rank functional variants without task-specific training.

Table 3: Zero-shot performance of JEPA-DNA using the DNABERT-2 backbone compared to the original DNABERT-2 baseline. Bold values denote superior zero-shot performance.

Task	DNABERT-2		JEPA-DNA (Ours)		Gain
	AUROC	AUPRC	AUROC	AUPRC	% AUROC
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%

Table 3 summarizes these results using DNABERT-2 as the representative backbone. 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, we observe architectural variability in task-specific gains; comprehensive zero-shot results for NTv3 and HyenaDNA are provided in Appendix Table 7.

4.4 Ablation Studies

Table 4: Evaluation of incremental impact of continual training with key components of the JEPA-DNA paradigm. Promoter, Splice, and Express denote GUE Promoter, GUE Splice Site, and BEND Expression Effect tasks, respectively. Bold values indicate the optimal configuration.

Configuration	**Components**				**AUROC**			**Avg. Gain**
	MLM	JEPA	VICReg	Sched. Mask	Promoter	Splice	Express.	% AUROC
Baseline (DNABERT-2)	×	×	×	×	0.884	0.653	0.543	—
+MLM	✓	×	×	×	0.929	0.674	0.532	+2.06%
+MLM+JEPA	✓	✓	×	×	0.927	0.669	0.529	+1.58%
+MLM+JEPA+SM	✓	✓	×	✓	0.925	0.687	0.519	+1.81%
+MLM+JEPA+VR	✓	✓	✓	×	0.928	0.689	0.565	+4.79%
+MLM+JEPA+VR+SM	✓	✓	✓	✓	**0.928**	**0.705**	**0.574**	**+6.23%**

Table 5: Evaluation of predictor head's architecture and depth. Bold values denote the best-performing configuration.

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%

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 4 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 Scheduled Masking (SM).

We further evaluate our *architectural and optimization choices*. Success in latent alignment depends on the predictor architecture; Table 5 shows that 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, further validating this design choice through embedding analysis detailed in the Appendix A.1. 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 9).

Finally, we compare *latent-space prediction against reconstruction-based paradigms* for continual training. Table 10 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. In addition, a comparison to a contrastive learning approach (DNASimCLR [14]) is provided in Appendix A.1, where Table 11 shows that JEPA-DNA consistently improves performance across all evaluated tasks.

5 Limitations and Future Work

While our results demonstrate that JEPA-DNA effectively enhances the representational quality of existing genomic foundation models, several avenues for further exploration remain. 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. A logical next step is to 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, while we utilized a representative JEPA architecture, integrating recent advancements such as LeJEPA [28], which introduces more efficient

prediction strategies, could potentially offer better scaling properties for the extremely long-range dependencies inherent in genomic data.

6 Conclusion

In this work, we introduced JEPA-DNA, a versatile framework that adapts the Joint Embedding Predictive Architecture for genomic sequence modeling. By shifting the focus from token-level reconstruction to the prediction of latent representations, we provide a method that captures high-level biological context more effectively than traditional generative objectives alone. Our results demonstrate that JEPA-DNA consistently improves the performance of established backbones across both supervised downstream tasks and zero-shot evaluations. This work paves the way toward AI understanding Nature through biologically meaningful representation learning.

References

[1] Gonzalo Benegas, Chengzhong Ye, Carlos Albors, Jianan Canal Li, and Yun S Song. Genomic language models: opportunities and challenges. *Trends in Genetics*, 2025.

[2] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

[3] Lorenzo Dalla-Torre, Nicolás Benegas, Daria Grechishnikova, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, 2023.

[4] 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.

[5] 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.

[6] Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 2022.

[7] 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*, 2026.

[8] 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*, 2025.

[9] Yu Ji, Zhiqiang Zhou, Han Liu, and Ramana V Davuluri. DNABERT: a comprehensive predictor for DNA sequences based on deep transfer learning. *Bioinformatics*, 37(24):4776–4783, 2021.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[11] 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*, pages 1–13, 2026.

[12] 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.

[13] Gonzalo Benegas, Carlos Albors, Alan J Aw, Chengzhong Ye, and Yun S Song. GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction. *bioRxiv*, 2024.

[14] 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.

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023.

[16] Hai Huang, Yann LeCun, and Randall Balestriero. LLM-JEPA: Large language models meet joint embedding predictive architectures. *arXiv preprint arXiv:2509.14252*, 2025.

[17] Elon Litman, Tyler Myers, Vinayak Agarwal, Ekansh Mittal, Orion Li, Ashwin Gopinath, and Timothy Kassis. GeneJepa: A predictive world model of the transcriptome. *bioRxiv*, 2025.

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

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

[20] Evan Trop, Yair Schiff, Edgar Mariano Marroquin, Chia Hsiang Kao, Aaron Gokaslan, McKinley Polen, Mingyi Shao, Aymen Kallala, Bernardo P de Almeida, Thomas PIERROT, Yang I Li, and Volodymyr Kuleshov. The genomics long-range benchmark: Advancing DNA language models, 2025.

[21] Frederikke Isa Marin, Felix Teufel, Marc Horlacher, Dennis Madsen, Dennis Pultz, Ole Winther, and Wouter Boomsma. BEND: Benchmarking DNA language models on biologically meaningful tasks. *arXiv preprint arXiv:2311.12570*, 2023.

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

[23] 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.

[24] 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, 2025.

[25] 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.

[26] 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.

[27] Tony Zeng and Yang I Li. Predicting RNA splicing from DNA sequence using pangolin. *Genome biology*, 23(1):103, 2022.

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

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

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

[31] Juntang Zhuang et al. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *NeurIPS*, 2020.

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

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

[34] Fiona Cunningham, James E. Allen, Jamie Allen, Jorge Alvarez-Jarreta, M. Ridwan Amode, Irina M. Armean, Olanrewaju Austine-Orimoloye, Andrey G. Azov, If Barnes, Ruth Bennett, et al. Ensembl 2024. *Nucleic Acids Research*, 52(D1):D891–D898, 2024.

[35] 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.

[36] Deanna M Church, Leo Goodstadt, LaDeana W Hillier, Michael C Zody, Steve Goldstein, Xingyi She, Carol J Bult, Richa Agarwala, J Michael Cherry, Michael DiCuccio, et al. Modernizing reference genome assemblies. *PLoS Biology*, 9(7):e1001091, 2011.

[37] 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.

[38] 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.

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

[40] The Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *Nature*, 408(6814):796–815, 2000.

[41] 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.

[42] 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*, 2024.

A Appendix

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

Section A.1 offers extended experimental results, including robustness (multi-seed) analysis, detailed performance breakdowns, and additional ablations.

Section A.2 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 Extended Results

Robustness Analysis. We conduct a multi-seed analysis using the DNABERT-2 backbone to evaluate the stability of the reported performance gains. As shown in Table 6, JEPA-DNA yields consistent and stable improvements.

Zero-Shot Performance for Additional Backbones. Table 7 provides the full zero-shot results for DNABERT-2 (Transformer + MLM + CLS; Multi-Species Dataset), Nucleotide Transformer v3 (Hybrid + MLM + Mean Pooling; Open-Genome 2 Dataset), and HyenaDNA (Long Convolutions + NTP + last-token; Reference-Genome Dataset), 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.

Table 6: Impact of JEPA-DNA on Representation Stability and Performance. We compare the DNABERT-2 baseline against its JEPA-DNA enhanced counterpart using linear probing across five random seeds. We report the mean AUROC and AUPRC with standard deviations. JEPA-DNA consistently improves both performance and stability (lower variance) across the majority of tasks, with relative AUROC gains of up to 7.8%. Bold values indicate the superior mean performance.

Task	**DNABERT-2 (Baseline)**		**JEPA-DNA (Ours)**		**Gain**
	AUROC	AUPRC	AUROC	AUPRC	% AUROC
GUE					
TF Binding	0.787 ± .001	0.773 ± .001	**0.839** ± .001	**0.828** ± .001	+6.61%
Promoter	0.885 ± .001	0.877 ± .001	**0.928** ± .000	**0.913** ± .000	+4.86%
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%
Expression Effect	0.673 ± .002	0.646 ± .001	**0.688** ± .001	**0.656** ± .001	+2.23%
Common vs. Rare	**0.508** ± .004	0.504 ± .006	0.505 ± .004	**0.505** ± .003	-0.59%
meQTL	0.619 ± .005	0.551 ± .004	**0.645** ± .004	**0.581** ± .005	+4.20%
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%

Baseline Performance Breakdown. To provide a comprehensive comparison against the state-of-the-art, Table 8 details the per-task linear probing performance of each individual baseline GFM (for a set of leading GFM architectures).

Extended Ablation Studies. Tables 9 and 10 present the detailed results for our ablation studies on optimization algorithms and training paradigms.

DNASimCLR Comparison. We compare our JEPA-DNA approach against DNASimCLR [14], a contrastive learning framework based on SimCLR [33]. Due to the absence of publicly available code, we reimplement the method following the original description and refine the official SimCLR

Table 7: We evaluate the zero-shot capabilities of JEPA-DNA augmented models against their original baselines. Bold values denote superior zero-shot performance within each backbone group.

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 8: Performance 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**	NA	0.550	0.572
LRB Causal eQTL	0.694	**0.697**	0.672	0.690	0.662	0.565	0.665

implementation accordingly, adapting it to the DNA domain. Specifically, 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 9: 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.

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 [29]	0.776	0.777	0.523	0.351	0.518	0.080	-12.24%
AdamW [30]	0.799	0.795	0.589	0.388	0.576	0.091	-4.45%
AdaBelief [31]	0.901	0.894	0.662	0.454	0.545	0.086	+1.22%
SGDM [32]	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+6.22%**

Table 10: 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.

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	+2.06%
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	+4.79%
JEPA	**MLM**	✓	✓	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+6.23%**

Table 11: DNASimCLR vs. JEPA-DNA across genomic tasks.

Learning Paradigm	**Promoter**		**Splice Site**		**Expression**		**Avg. Gain**
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	% AUROC
DNASimCLR [14]	0.922	0.912	0.600	0.407	0.549	0.085	—
JEPA-DNA (Ours)	**0.928**	**0.913**	**0.705**	**0.500**	**0.574**	**0.091**	**+7.57%**

Table 11, 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%.

Embeddings Analysis. We defined a set of genomic loci on human chromosome 22 (GRCh38) using public Ensembl annotations [34], 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 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-neighbour classifier applied in the original embedding space with stratified 5-fold 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.

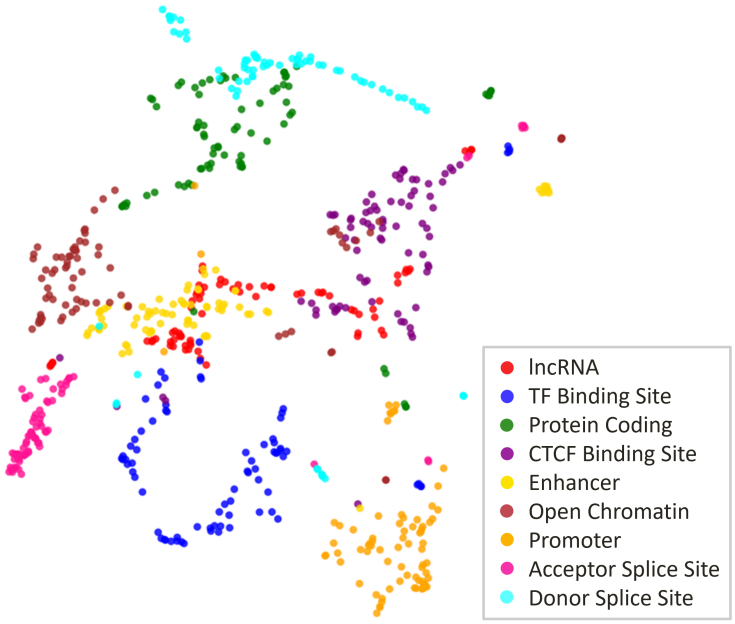

Figure 2: JEPA-DNA Embedding visualization colored by Ensembl annotations.

A.2 Extended Implementation Details

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

- **DNABERT-2 [2]:** 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 [8]:** 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 [4]:** 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.

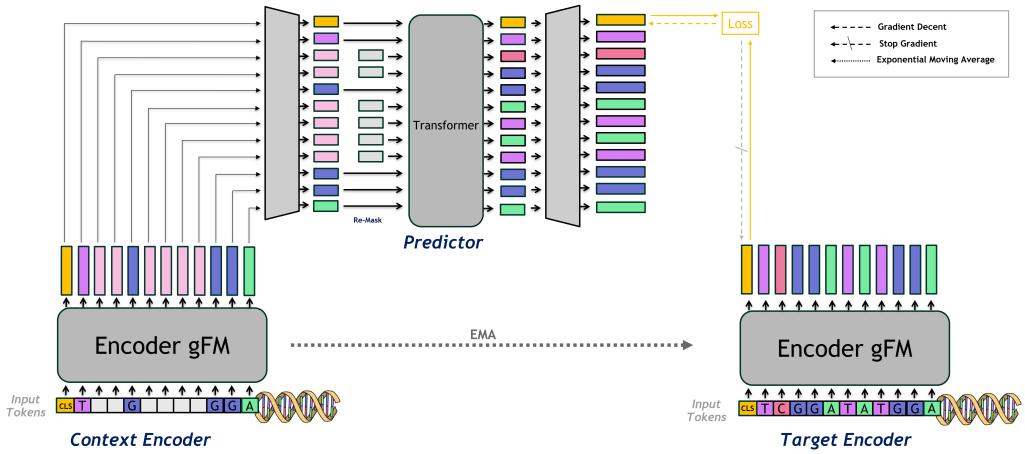

Figure 3: The JEPA-DNA architecture using a CLS-based representation, as implemented with DNABERT-2.

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) [35] together with genomic data from five model organisms: mouse [36], zebrafish [37], fruit fly [38], nematode [39], and thale cress [40]. All genome sequences are obtained from UCSC genome Browser [41], 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 [11], 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 [35], 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 [7]. Our supervised evaluation includes three standard classification tasks from the GUE benchmark [2]: promoter prediction, transcription factor binding site prediction, and splice site prediction. Additionally, we utilize variant effect prediction tasks from VariantBenchmarks [19], 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) [20], 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 [21] for expression and disease-associated variants, and TraitGym [22] for predicting Mendelian and complex traits. Furthermore, we evaluate clinical pathogenicity prediction using the Song-Lab ClinVar dataset [24] and the non-coding pathogenic OMIM task from LRB [20]. In addition, we include variant-effect prediction on BRCA1 saturation mutagenesis [23], and a LOL-EVE causal eQTL task [42], 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) [32] 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 early stopping based on performance on the GUE Splice Site validation set. 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.