Janus: An Efficient and Expressive Subquadratic Architecture for Modeling Biological Sequences

Krithik Ramesh [*12] **Sameed Siddiqui** [*13] **Michael Mitzenmacher** [14] **Pardis C. Sabeti** [156]

Abstract

Deep learning tools such as convolutional neural networks (CNNs) and transformers have spurred great advancements in computational biology. However, existing methods are constrained architecturally in context length, computational complexity, and model size. This paper introduces Janus, a sub-quadratic architecture for sequence modeling, which combines projected gated convolutions and structured state spaces to achieve local and global context with single-nucleotide resolution. Janus outperforms CNN-, GPT-, BERT-, and long convolution-based models in many tested genomics tasks without pre-training and with 4x-781x fewer parameters. In the proteomics domain, Janus similarly outperforms pre-trained attention-based models, including ESM-1B and TAPE-BERT, on remote homology prediction without pre-training and while using 3,308x-23,636x fewer parameters. Janus couples these performance improvements with reduced wall-clock times, showing up to 50x speed-up compared to ESM1b and 7x speed-up compared to DistilProtBert for sequences with length up to 16,384.

1. Introduction

Increasingly sophisticated deep learning models are used to understand biological systems, with emergent work relying on larger pre-trained models to capture the underlying sequence-function relationships hidden in the genomic and proteomic landscapes. While these techniques have shown promise, they still possess inherent limitations that hinder efficient modeling of sequences at scale, a challenge particularly relevant in fields such as genomics with large datasets and complex chemical relationships between sequences.

In computational biology, two architectural paradigms have dominated: convolutional neural networks (CNNs) and transformers, each with distinct strengths and limitations. CNNs excel in detecting localized patterns, such as DNA motifs (Zhou & Troyanskaya, 2015; Xiang et al., 2021), through highly parallelizable convolutions. However, their fixed-length kernels constrain their receptive field, hindering the capture of long-range relationships even with multiple filters and dilated convolutions (Avsec et al., 2021). Conversely, transformers adeptly model global pairwise relationships, showcasing remarkable performance in generative and classification tasks (Li et al., 2023; Avsec et al., 2021), but are hampered by the quadratic complexity of their attention mechanisms.

The ideal architecture for biological sequence analysis must efficiently integrate both local and global contexts to address the complex interplay of short- and long-range interactions. Recognizing this need, recent efforts have focused on developing models that balance these aspects. Notable examples include the State Spaces Sequence-to-Sequence model (S4) and Hyena (Gu et al., 2021a; Poli et al., 2023), which use convolutions to enhance state space modeling, combining them with multi-layer perceptrons to create dynamic, input-dependent long convolution kernels.

Building upon these insights, we introduce **Janus**, a novel architecture that combines projected gated convolutions with S4D layers to achieve both gating and input-dependent filtering. Janus extends the data modulation concept of BaseConv (Arora et al., 2023) by incorporating learnable linear projections and root mean square normalization (RMSNorm) (Zhang & Sennrich, 2019) before and after a depth-wise 1D convolution. This design enables efficient extraction of local sequence features, while a parallel linear projection captures global context. The element-wise product of these local and global features, followed by projection into an S4D layer, allows Janus to comprehensively analyze sequences, ac-

[*] Equal contribution [1] Broad Institute of MIT and Harvard, Cambridge, MA, USA [2] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA [3] Computational and Systems Biology PhD Program, MIT, Cambridge, MA, USA [4] School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA [5] Department of Organismic and Evolutionary biology, Harvard University, Cambridge, MA, USA [6] Howard Hughes Medical Institute, Chevy Chase, MD, USA. Correspondence to: Krithik Ramesh <krithik@mit.edu>, Pardis C Sabeti <pardis@broadinstitute.org>.

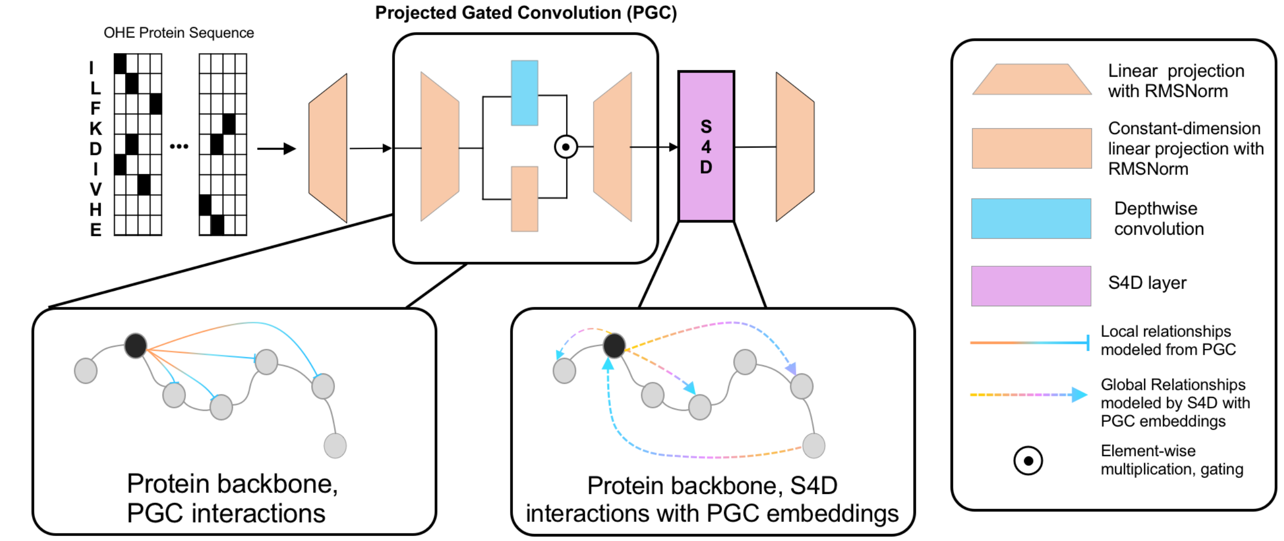
Projected Gated Convolution (PGC)
OHE Protein Sequence
I
L
F
K
D
I
V
H
E
S4D
Linear projection with RMSNorm
Constant-dimension linear projection with RMSNorm
Depthwise convolution
S4D layer
Local relationships modeled from PGC
Global Relationships modeled by S4D with PGC embeddings
Element-wise multiplication, gating
Protein backbone, PGC interactions
Protein backbone, S4D interactions with PGC embeddings

Figure 1. Overview of Janus applied to protein sequence analysis. The architecture employs a Projected Gated Convolution (PGC) to encode one-hot encoded (OHE) protein sequences into a rich feature representation, capturing local interaction patterns within the protein backbone. These PGC embeddings are further processed through an S4D layer, which integrates both local and global sequence information. The model effectively combines local structural insights with global contextual relationships, enabling accurate prediction of protein properties.

counting for both short- and long-range dependencies. The result is a model that emulates the sub-sequence interaction capabilities of transformers with enhanced efficiency and scalability—a crucial advantage for biological tasks involving long, complex sequences. By leveraging input-dependent gating and convolution filters, Janus strikes a balance between the expressiveness of attention mechanisms and the efficiency of convolutions, potentially establishing a new benchmark for sequence modeling in computational biology.

We evaluate Janus on a broad array of biological tasks, achieving state-of-the-art (SOTA) performance in most tasks while using significantly fewer parameters than competing models. Across chromatin profiling, gene regulation, and clustered regularly interspaced short palindromic repeats (CRISPR)- related tasks, Janus outperforms CNN-, BERT-, GPT-, and long convolution-based models while using 4-30x fewer parameters. In protein-related sequence modelling tasks, a 55,000 parameter Janus model outperformed models ESM-1B (Rives et al., 2021) and TAPE-BERT (Rao et al., 2019), which are 650 million and 91 million parameter pretrained models, respectively, using a 55,000 parameter Janus model without pre-training.

We highlight three main contributions of this work.

- First, we introduce a new model architecture, Janus, that is highly expressive, lightweight, and straightforward to implement.
- Second, this study demonstrates the broadest application of efficient convolutions and state spaces to biological tasks.
- Third, by outperforming existing state of the art models with significantly smaller Janus models, our model establishes a new promising subfamily of compact and easy to implement subquadratic architectures.

2. Preliminaries and Related Work

2.1. CNNs, Transformers, and Hybrid Approaches in Biological Sequence Modeling

Convolutional Neural Networks (CNNs) have demonstrated robust performance in biological sequence modeling tasks, from CRISPR enzyme activity prediction to DNA architecture prediction. CNNs excel in capturing local patterns such as DNA-binding motifs, leveraging their high parallelizability and shift equivariance (Zhou & Troyanskaya, 2015; Xiang et al., 2021). However, their fixed-length kernels constrain their receptive field, limiting their ability to model long-range interactions crucial in many biological processes (Avsec et al., 2021).

Transformers, with their attention mechanism, address this limitation by computing pairwise interactions across all positions in a sequence. This global context modeling has proven particularly effective for tasks involving long-range dependencies, such as protein folding and enhancer-promoter interactions (Jumper et al., 2021; Avsec et al., 2021). The attention mechanism, defined as:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V are query, key, and value projections of the input, enables controlled, gated data flow. Despite their success in models like AlphaFold and ESM-Fold (Jumper et al., 2021; Rives et al., 2021), transformers struggle with capturing local motifs and face quadratic computational complexity with sequence length, limiting their application to very long sequences.

Recognizing the complementary strengths of CNNs and Transformers, hybrid approaches have emerged to leverage both local and global contexts. Models like ProteinBERT employ CNNs for input sequences and linear layers for annotations, feeding their outputs into a global attention layer (Brandes et al., 2022). Hierarchical architectures, such as Shifting Window Attention (Swin) transformers, use different attention blocks across various context lengths to capture multi-resolution interactions (Li et al., 2023). Another approach, exemplified by ProtFlash, combines quadratic attention for local chunks with linear attention for global context (Wang et al., 2023). While these hybrid models have shown promising results, they still grapple with computational efficiency, especially as sequence lengths approach genomic scales.

2.2. Structured State Space Models and Gating Strategies

Structured state space models (SSMs) have emerged as a promising approach to address the limitations of transformers and convolutions in long-range sequence modeling (Gu et al., 2021a;b). These models, such as S4 and Mamba (Gu et al., 2021a; Gu & Dao, 2023), efficiently approximate and memorize long sequences by leveraging a linear state space representation:

$$\begin{aligned} x'(t) &= \boldsymbol{A}(t)x(t) + \boldsymbol{B}(t)u(t) \\ y(t) &= \boldsymbol{C}(t)x(t) \end{aligned} \tag{2}$$

where $u(t)$, $x(t)$, and $y(t)$ represent the input, latent state, and output, respectively. The efficiency of SSMs stems from their ability to dynamically represent sequences through learnable parameters $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ (Gu et al., 2021a).

S4D (Gu et al., 2022), a variant of S4, further enhances efficiency through state space diagonalization. It computes an implicit convolutional kernel $K(t)$ that captures temporal dynamics:

$$K(t) = \boldsymbol{C}e^{t\boldsymbol{A}}\boldsymbol{B}, \quad y(t) = (K * u)(t) \tag{3}$$

This formulation enables S4D to have parallelized training, making it particularly suitable for modeling long biological sequences.

Complementing these advances, recent work has focused on enhancing the expressiveness of convolution-based models through gating and data modulation strategies. Models like H3 (Fu et al., 2022) and Hyena (Poli et al., 2023) incorporate attention-like gating mechanisms or dense activations to achieve data-dependent modulation similar to transformer attention.

A key insight from recent studies on associative recall tasks (Arora et al., 2023) reveals that gated convolutions can effectively solve these tasks when the hidden dimension of the convolution is $\log_2(\text{sequence length})$ and gated with a linear layer. This finding, exemplified in the BaseConv architecture, extends convolutions with input-dependent mixing, allowing for efficient evaluation of subsequence interactions while maintaining sub-quadratic computational complexity.

3. Methods

The Janus architecture integrates two distinct stages for enhanced sequence processing: (1) first, a projected gated convolution module, which builds upon the BaseConv(Arora et al., 2023) model of Arora *et al.* by incorporating linear projections coupled with RMSNorm at the input, gating, and output stages; and (2) next, a second stage diagonalized state space model, S4D, which leverages the mixed input tokens from the first stage. This setup facilitates the learning of both local and global context within sequences, capitalizing on the strengths of the S4D architecture to address complex dependencies in the data.

3.1. Projected BaseConv Module

At the first stage of our model, a projected biological sequence represented by $\boldsymbol{u} \in \mathbb{R}^{N \times d}$, where N is the sequence length and d is the projected feature dimensionality, undergoes two primary transformations. First, in each layer ℓ the sequence $\boldsymbol{u}$ is linearly projected using a weight matrix $\boldsymbol{W}_{in}^{\ell} \in \mathbb{R}^{d \times d'}$ and a bias vector $\boldsymbol{b}_{in}^{\ell} \in \mathbb{R}^{N \times d'}$, where d' is the internal projection dimension. This linear projection, followed by RMS normalization, transforms the sequence to emphasize its global context. This output $\boldsymbol{u}'_{proj}$ is then processed through a depthwise 1D convolutional layer, applying a set of d' learnable filters $\boldsymbol{h}^{\ell} \in \mathbb{R}^{N'}$ to the sequence, where $N' < N$, and the addition of another bias vector $\boldsymbol{b}_2^{\ell} \in \mathbb{R}^{N \times d'}$. This convolution, adept at extracting local features, maintains shift equivariance, ensuring sensitivity to the relative positioning of features within $\boldsymbol{u}$ and capturing local dependencies. In parallel, a linear projection of $\boldsymbol{u}'_{proj}$ computes global features using a weight matrix $\boldsymbol{W}^{\ell} \in \mathbb{R}^{d' \times d'}$ and a bias vector $\boldsymbol{b}_1^{\ell} \in \mathbb{R}^{N \times d'}$. The resulting vectors of the convolution and projection of $\boldsymbol{u}'_{proj}$ are then element-wise multiplied to form $\boldsymbol{u}'_{conv}$. This is then further mixed via a subsequent projection using weight matrix $\boldsymbol{W}^{\ell}_{out} \in \mathbb{R}^{d' \times d}$ and a bias vector $\boldsymbol{b}^{\ell}_{out} \in \mathbb{R}^{N \times d}$, and followed by an RMS normalization step. This process ensures a thorough integration of both local and global features, crucial for effective modeling of biological sequences.

The projected BaseConv Module can be formulated as:

$$\boldsymbol{u'}_{\text{proj}} = \text{RMSNorm}\left(\boldsymbol{u} \cdot \boldsymbol{W}^{\ell}_{\text{in}} + \boldsymbol{b}^{\ell}_{\text{in}}\right) \quad (4)$$

$$\boldsymbol{u'}_{\text{conv}} = \text{RMSNorm}\left(\boldsymbol{u'}_{\text{proj}} \cdot \boldsymbol{W}^{\ell} + \boldsymbol{b}^{\ell}_{1}\right) \odot \left(\boldsymbol{h}^{\ell} * \boldsymbol{u'}_{\text{proj}} + \boldsymbol{b}^{\ell}_{2}\right) \quad (5)$$

$$\boldsymbol{y'} = \text{RMSNorm}\left(\boldsymbol{u'}_{\text{conv}} \cdot \boldsymbol{W}^{\ell}_{\text{out}} + \boldsymbol{b}^{\ell}_{\text{out}}\right) \quad (6)$$

3.2. S4D for Long Range Sequence Modeling

A key insight of our work is to use our first stage projected and gated convolution results as an input to a structured state space model with diagonalized state spaces (S4D) (Gu et al., 2022). As such, the combined architecture leverages both local and global context provided by the first stage to enhance its capacity for modeling long-range dependencies. The Janus model outputs, enriched by the gating mechanism, are projected back to the hidden state size compatible with S4D. This integration allows S4D to operate on a more expressive latent space informed by the nuanced representations captured by projected BaseConv.

By integrating the outputs from the projected BaseConv into S4D, Janus benefits from the first stage's ability to capture both local and global context. This enhanced representation, fed into the S4D model, allows for a more comprehensive understanding of the sequence dynamics. The structural basis functions of S4D effectively process these enriched representations, enabling the model to capture complex, long-range dependencies inherent in sequential data. This integration not only boosts the expressive power of the latent space but also ensures that the model is well-equipped to handle the intricacies of various tasks, be it classification or regression, in the realm of computational biology.

3.3. Biological Domain Explorations

To benchmark performance and generalization, we evaluate Janus across diverse biological prediction tasks without any pretraining. These encompass major genomic and proteomic challenges including chromatin profiles, gene regulation, CRISPR activity, and protein fitness landscapes. This selection tests intrinsic model capacity to tackle distinct learning objectives pertinent to key areas of computational biology.

4. Experiments

We assess Janus on tasks spanning major biological domains without specialized tuning or pretraining. In genomics, we predict chromatin profiling of DNA sequence(Zhou & Troyanskaya, 2015) and performance in gene regulation on the GenomicBenchmark (Grešová et al., 2023) dateset. We also predict CRISPR editing efficacy (Metsky et al., 2022; DeWeirdt et al., 2022) and in proteomics, we model fitness landscapes (Castro et al., 2022), enzymatic activities, and complex structural properties using the Tasks Assessing Protein Embeddings (TAPE) dataset (Rao et al., 2019). We compare off-the-shelf Janus performance to state-of-the-art models to elucidate the tradeoffs between specialized inductive biases and generalization capacity. This comprehensive evaluation probes intrinsic versatility to tackle varied regression and classification objectives with Janus.

4.1. Genomics tasks

Chromatin profiling: Given the pivotal role of epigenetic regulatory activity in controlling gene expression, we next tested Janus in this domain. The DeepSEA dataset (Zhou & Troyanskaya, 2015) is employed for this evaluation, as it extensively profiles human genomic epigenetic regulatory activity using DNase-seq and ChIP-seq assays. This dataset annotates 919 chromatin accessibility and histone modification features at single nucleotide resolution, posing a 919-way multilabel classification challenge essential for evaluating a model's capacity to decode the regulatory DNA language and comprehend long-range chromosomal grammar. In tests involving 1,000 nucleotides long genomics sequences (Table 2), a Janus model with 678k parameters achieves an SOTA AUC-ROC of 93.1 on DNase I-hypersensitive sites (DHS). However, we note that while Janus performs competitively with competing models with a 1,000 sequence length, there is a persistent 3-4% performance gap for histone mark classification compared to models evaluated on sequences of length 8,000.

GenomicBenchmarks: In a standardized suite of genomics benchmarks, which includes a variety of classification tasks targeting key gene-regulating regions (Table 1), the Janus model achieves notably better performance against SOTA baselines, despite being significantly more compact. Janus is approximately four times smaller than any other model in this comparison, yet it consistently surpasses larger models. These benchmarks evaluate Janus's ability to process sequences ranging from 200 to 4,776 bases. Remarkably, without any pre-training, Janus outperforms the pre-trained DNABERT (Ji et al., 2021) in 7 of 8 tasks. It also exceeds the performance of a pre-trained GPT-based DNA model in 6 out of 8 tasks, with equal performance in another task. When compared to the long convolution-based HyenaDNA, Janus demonstrates superior results in 7 out

Table 1. Model performance on GenomicBenchmark Datasets on Top-1 (%) accuracy

Models	Janus (ours)	Gpt	HyenaDna	HyenaDNA	DNABERT
Pretrained	no	yes	no	yes	yes
Model Parameters	106K	529k	436k	436k	110M
Mouse Enhancers	80.9	79.3	84.7	**85.1**	66.9
Coding vs Intergenomic	**94.0**	91.2	90.9	91.3	92.5
Human vs Worm	**96.6**	**96.6**	96.4	**96.6**	96.5
Human Enhancers Cohn	73.4	72.9	72.9	**74.2**	74.0
Human Enhancers Ensembl	86.8	88.3	85.7	**89.2**	85.7
Human Regulatory	93.3	91.8	90.4	**93.8**	88.1
Human Nontata Promoters	**96.7**	90.1	93.3	96.6	85.6
Human OCR Ensemble	79.9	79.9	78.8	**80.9**	75.1

Table 2. Comparative Analysis on Chromatin Profile 919-way classification: AUC-ROC for prediction in transcription factor (TF), DNase I-hypersensitive sites (DHS), and histone markers (HM)

				AUC-ROC	
Model	Params	Len	TF	DHS	HM
DeepSEA	40M	1k	95.8	92.3	85.6
BigBird	110M	8k	96.1	92.1	88.7
HyenaDNA	7M	1k	**96.4**	93.0	86.3
HyenaDNA	3.5M	8k	95.5	91.7	**89.3**
Janus	678K	1k	95.9	**93.1**	86.1

of 8 tasks when both models are not pre-trained. Even in scenarios where HyenaDNA is pre-trained and Janus is not, Janus still outperforms HyenaDNA in 3 out of 8 tasks. This highlights Janus's efficiency and robustness, especially notable given its significantly smaller size and ability to handle complex genomic sequences without extensive pre-training.

4.2. CRISPR Tasks

In CRISPR technologies, we rigorously evaluate Janus models across two applications: viral diagnostics using Cas13 and gene edit targeting with Cas9. CRISPR enzymes can be programmed using a "guide" RNA sequence to find and respond to a specific target sequence, with the strength of response differeing with respect to the specific guide-target sequence pair.

Cas13 diagnostics: We find that Janus demonstrates SOTA performance in Cas13-related tasks (Table 3) with 31.6x fewer parameters than the CNN-based ADAPT model. Specifically, in classification tasks, Janus has an AUC-ROC and AUPR of 0.939 and 0.990, respectively, compared to 0.866 and 0.972 for the ADAPT model. In regression tasks, Janus again outperforms the CNN-based model, with Spearman's correlation coefficients of 0.856 and 0.810, compared to 0.774 and 0.686 for the ADAPT models looking at all guide-target pairs and only positive-identified guide-target pairs, respectively. Highlighting the efficiency and expressivity of Janus, these performance gains were achieved with a model comprising only 3.8k parameters, in contrast to the ADAPT model's 120k parameters.

Cas9 genome editing: Janus exhibits similarly promising performance in the Cas9 genome editing domain, beating pre-established models for Cas9 performance in almost all tested datasets. Across all 9 tested datasets (Table 4), Janus achieves an average Spearman's correlation of 0.51, compared to 0.45 and 0.36 for CRISPRon(Xiang et al., 2021) and DeepSpCas9 (Kim et al., 2019), both highly-used CNN-based models. Impressively, in the Behan2019 dataset, Janus more than doubled the correlation score of CRISPRon (Xiang et al., 2021) and DeepSpCas9 (Kim et al., 2019), with a coefficient of 0.439 compared to 0.219 and 0.198, respectively.

Table 3. Comparative Analysis on Cas13a: AUC-ROC, AUPR, Spearman's Correlations, and Model Parameters

	Adapt CNN	Janus (Ours)
Model Parameters	120k	3.8k
AUC-ROC	0.866	**0.939**
AUPR	0.972	**0.990**
All Guide-Targets Spearman's	0.774	**0.856**
Positive Only Spearman's	0.686	**0.810**

Table 4. Comparative Analysis on Cas9: 5-fold Spearman's Correlations, and Model Parameters

Dataset	Janus (Ours)	CRISPRon	DeepSpCas9
Model Parameters	13.3k	420k	320k
Doench2014_mouse	**0.508**	0.445	0.432
Doench2014_human	**0.513**	0.457	0.454
Doench2016	**0.416**	0.386	0.389
Wang2014	**0.421**	0.359	0.050
Munoz2016	**0.474**	0.317	0.085
Behan2019	**0.439**	0.219	0.198
Kim2019	0.747	**0.896**	0.773
Aguirre216	**0.562**	0.538	0.525

4.3. Protein Tasks

Proteins are complex biomolecules whose sequence directly determines structure and function. A key challenge is modeling higher-order epistatic effects, wherein amino acids interact nonlinearly and at varying distances to alter protein properties (Cadet et al., 2022). As such, protein-related tasks serve as ideal tests for the Janus architecture, which was specifically designed to evaluate interactions at varying distances.

Protein Fitness: We first test Janus on a group of three protein datasets exhibiting epistasis: the Gifford antibody enrichment dataset, which shows sequence viability over selection rounds; the GB1 dataset, which combines stability and binding affinity to define fitness across a mutational landscape; and the GFP fluorescence dataset, which directly quantifies mutant functionality. Each dataset consists of protein sequences ranging in length from 20 to 237 amino acids as inputs and either log_fluoroence or CRD3 enrichment regression targets. We compare our model against the SOTA Regularized Latent Space Optimization (ReLSO) model (Castro et al., 2022) which is comprised of a series of 10 transformer encoder layers and 4 decoding heads that simultaneously predict the protein sequence and assess the fitness of the encoded embeddings derived from the sequence. In these tests (Table 5), Janus outperforms three ReLSO variants on all three datasets, and surpasses the other two variants (ReLSO-Interp and ReLSO-$\alpha = 0.5$) in two datasets while matching performance on a third dataset. Notably, Janus achieves these SOTA performances with a model size of 55,000 parameters, compared to the 7-8.3 million parameters in the ReLSO decoder blocks alone.

Table 5. Spearman correlation scores for different models on protein fitness datasets for antibody binding (Gifford dataset), antibody fitness (GB1 dataset), and green fluorescent protein (GFP) brightness

MODEL	GIFFORD (AB BINDING)	GB1 (AB FITNESS)	GFP
RELSO (INTERP)	0.48	0.43	**0.86**
RELSO (NEG)	0.47	0.42	0.77
RELSO $\alpha = 0.1$	0.35	0.53	0.84
RELSO $\alpha = 0.5$	**0.50**	0.45	0.85
RELSO	0.48	0.44	0.70
JANUS (OURS)	0.49	**0.61**	**0.86**

TAPE Protein Benchmarks: We next test Janus against a larger family of attention-based protein models across Tasks Assessing Protein Embeddings (TAPE) (Rao et al., 2019), (Table 6) a well-established suite of proteomic benchmarking datasets. Specifically, we evaluate our model on predicting remote homology, fluorescence, and protein stability. We compete against DistilProteinBert (Geffen et al., 2022) (230M parameters), ESM-1b (Rives et al., 2021) (650M parameters), ProtFlash (Wang et al., 2023) (174M parameters) — all models that have been pre-trained on millions of protein sequences from pFam (Mistry et al., 2021) and Uniref90 (Suzek et al., 2015). Janus achieves SOTA performance on two out of the three (fluorescence and super-family top-1 remote homology) benchmarks with a 55,000 parameters model without pretraining — reducing parameter count by up to 11,818x while increasing performance. Although Janus reached SOTA performance in two out of three tasks, it struggled on the stability regression task. We determined that this was due to overfitting, which was still present in smaller Janus models with as few as 4,000 parameters.

Table 6. Model Performance on TAPE Datasets; including fluorescence prediction (fluor), protein stability prediction, and remote homology super-family (RH)

MODEL	# PARAMS	FLUOR	STABILITY	RH
TAPE-BERT	91M	0.64	0.73	0.34
DISTILPROTBERT	230M	0.67	0.74	0.52
ESM-1B	650M	0.47	0.77	0.50
PROTFLASH-BASE	174M	**0.68**	**0.79**	0.50
JANUS (OURS)	55K	0.62	0.43	**0.59**

5. Discussion

Janus introduces a novel sequence modeling architecture that achieves state-of-the-art performance across diverse biological challenges while using significantly fewer parameters than established models. Its effectiveness arises from the combination of two key innovations: RMS-normalized projected gated convolutions and a diagonalized state space model (S4D). This architecture enables efficient mixing of local features and captures contextualized global interactions critical for modeling complex biochemical phenomena, all without pretraining requirements. Our comprehensive evaluation across genomics, CRISPR, and proteomics domains demonstrates Janus's exceptional generalizability and effectiveness, with particularly dramatic improvements in protein modeling tasks. This success in capturing both short- and long-distance interactions in protein sequences validates our architectural choices and underscores Janus's potential as a versatile, efficient solution for biological sequence modeling.

Despite these advances, Janus faces challenges in certain complex prediction tasks, such as histone mark classification, where it falls slightly short of some specialized models. These limitations, reminiscent of findings by Notin et al. (Notin et al., 2023) in proteomics, suggest areas for future exploration, including pretraining strategies and model scaling. We plan to investigate the efficacy of increased hidden size, layer count, and pretraining on complete genomic datasets to address these gaps. Furthermore, Janus's promising results open exciting avenues for integration into generative models like RFDiffusion (Watson et al., 2023) for advanced structure generation and protein design. By exploring Janus's scalability as a backbone for both score-based diffusion and autoregressive tasks, we aim to position it as a versatile alternative to traditional transformers in computational biology, potentially revolutionizing sequence modeling across the field.

References

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Ré, C. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*, 2023.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Cadet, F., Saavedra, E., Syren, P.-O., and Gontero, B. Machine learning, epistasis, and protein engineering: From sequence-structure-function relationships to regulation of metabolic pathways. *Frontiers in Molecular Biosciences*, 9:1098289, 2022.

Castro, E., Godavarthi, A., Rubinfien, J., Givechian, K., Bhaskar, D., and Krishnaswamy, S. Transformer-based protein generation with regularized latent space optimization. *Nature Machine Intelligence*, 4(10):840–851, 2022.

DeWeirdt, P. C., McGee, A. V., Zheng, F., Nwolah, I., Hegde, M., and Doench, J. G. Accounting for small variations in the tracrrna sequence improves sgrna activity predictions for crispr screening. *Nature Communications*, 13(1):5255, 2022.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Re, C. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations*, 2022.

Geffen, Y., Ofran, Y., and Unger, R. Distilprotbert: a distilled protein language model used to distinguish between real proteins and their randomly shuffled counterparts. *Bioinformatics*, 38(Supplement_2):ii95–ii98, 2022.

Grešová, K., Martinek, V., Čechák, D., Šimeček, P., and Alexiou, P. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021a.

Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., and Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585, 2021b.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983, 2022.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kim, H. K., Kim, Y., Lee, S., Min, S., Bae, J. Y., Choi, J. W., Park, J., Jung, D., Yoon, S., and Kim, H. H. Spcas9 activity prediction by deepspcas9, a deep learning–based model with high generalization performance. *Science advances*, 5(11):eaax9249, 2019.

Li, Z., Das, A., Beardall, W. A., Zhao, Y., and Stan, G.-B. Genomic interpreter: A hierarchical genomic deep neural network with 1d shifted window transformer. *arXiv preprint arXiv:2306.05143*, 2023.

Metsky, H. C., Welch, N. L., Pillai, P. P., Haradhvala, N. J., Rumker, L., Mantena, S., Zhang, Y. B., Yang, D. K., Ackerman, C. M., Weller, J., et al. Designing sensitive viral diagnostics with machine learning. *Nature biotechnology*, 40(7):1123–1131, 2022.

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L., Tosatto, S. C., Paladin, L., Raj, S., Richardson, L. J., et al. Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419, 2021.

Nguyen, E., Poli, M., Faizi, M., Thomas, A. W., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C. M., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Notin, P., Weitzman, R., Marks, D. S., and Gal, Y. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. *bioRxiv*, pp. 2023–12, 2023.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. *arXiv preprint arXiv:2302.10866*, 2023.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Wang, L., Zhang, H., Xu, W., Xue, Z., and Wang, Y. Deciphering the protein landscape with protflash, a lightweight language model. *Cell Reports Physical Science*, 4(10), 2023.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Xiang, X., Corsi, G. I., Anthon, C., Qu, K., Pan, X., Liang, X., Han, P., Dong, Z., Liu, L., Zhong, J., et al. Enhancing crispr-cas9 grna efficiency prediction by data integration and deep learning. *Nature communications*, 12(1):3238, 2021.

Zhang, B. and Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhou, J. and Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

A. Experimental Details

In the following section, we provide details the Janus model instantiation and training procedures for all tasks. All tasks were evaluated on Nvidia GPUs, either an A100-40GB or H100-80GB.

A.1. Genomic Tasks

A.1.1. CHROMATIN PROFILING

Table 7. Janus Model Configuration for Chromatin Profiling

PARAMETER	678,183
D_MODEL	256
N_LAYERS	2
DROPOUT	0.2
D_INPUT	4
D_OUTPUT	919
PRENORM	TRUE
PGC BLOCK 1	16 HIDDEN DIM, 0.2 DROPOUT
PGC BLOCK 2	128 HIDDEN DIM, 0.2 DROPOUT

Experiment Details: The DeepSEA dataset (Zhou & Troyanskaya, 2015) aggregated 919 attributes including 690 transcription factor (TF) binding profiles spanning 160 distinct TFs, alongside 125 DNase I hypersensitive sites (DHS) and 104 histone modification (HM) profiles. The dataset is constructed from 1,000 base pair sequences extracted from the hg19 human reference genome, with each sequence linked to a 919-dimensional target vector indicating the presence or absence of a chromatin feature peak within the central 200 base pairs. The adjacent 400 base pair regions provide extended context, crucial for accurate feature prediction. Strict non-overlapping training and testing sets are partitioned by chromosome, featuring 2.2 million training samples and 227. Each of these sequences was one-hot-encoded and trained using binary cross entropy loss with the AdamW optimizer with 0.001 learning rate and 0.01 weight decay. Janus was trained over 200 epochs, aligning with the methodology delineated in HyenaDNA (Nguyen et al., 2023) and evaluated the median AUC-ROC, for each of the 919 classes within the subset of DHS, TF, and HM profiles.

A.1.2. GENOMICSBENCHMARK

Table 8. Janus Model Configuration for GenomicBenchmark

PARAMETER	106,434
D_MODEL	128
N_LAYERS	1
DROPOUT	0.2
D_INPUT	4
D_OUTPUT	2
PRENORM	TRUE
PGC BLOCK 1	16 HIDDEN DIM, 0.2 DROPOUT
PGC BLOCK 2	128 HIDDEN DIM, 0.2 DROPOUT

Experimental Details: In our investigation utilizing the GenomicsBenchmark (Grešová et al., 2023) suite, we focused on eight binary classification tasks related to regulatory genomic elements. The datasets within this suite presented a diverse range of sequence lengths, varying from 200 to approximately 4800 base pairs. To standardize the input, we employed one-hot encoding for the sequences, padding them to the maximum length specific to each dataset. In cases of absent sequences, padding was implemented using the 'N' token, represented by [0,0,0,0]. Our training protocol involved a consistent 500 epochs for each dataset, optimizing the model with AdamW, a learning rate of 0.001, and a weight decay of 0.01, under the guidance of cross-entropy loss. We evaluated each dataset on top-1% accuracy metric for each dataset.

A.2. CRISPR Tasks

A.2.1. ADAPT CAS13

Table 9. Janus Model Configuration for Cas13a classification and regression tasks

PARAMETER	3,793 - 3,810
D_MODEL	16
N_LAYERS	1
DROPOUT	0.2
D_INPUT	8
D_OUTPUT	1,2
PRENORM	TRUE
PGC BLOCK 1	16 HIDDEN DIM, 0.2 DROPOUT

Experiment Details: For the CRISPR Cas13 dataset (Metsky et al., 2022), we encoded guide-target pairs using a one-hot encoding scheme with a dimensionality of 4 for each guide and target. These were then concatenated to form a stacked representation with an 8-dimensional one-hot-encoded vector for sequences of 48 base pairs. The log fluorsence threshold to distinguish active from non-active pairs was set at a value of -4.00. Our model underwent 5-fold cross-validation across three distinct tasks. In the first task, binary classification of guide-target pairs was performed, assessing the model's performance through AUC-ROC and AUPR metrics, with each fold being trained for 75 epochs. The following two tasks involved regression analyses: the first was a positive-only regression targeting values above the activity threshold, and the second encompassed a comprehensive regression across all guide-target pairs, both positive and negative. Both regression tasks were evaluated using Spearman's coefficient, following the same 75-epoch, 5-fold cross-validation structure.

A.2.2. CAS9

Table 10. Janus Model Configuration for Cas9 classification and regression tasks

PARAMETER	13,361
D_MODEL	48
N_LAYERS	1
DROPOUT	0.2
D_INPUT	4
D_OUTPUT	1
PRENORM	TRUE
PGC BLOCK 1	16 HIDDEN DIM, 0.2 DROPOUT

Experimental Details: We utilized a composite of seven CRISPR Cas9 datasets—Kim2019_train, Doench2014_mouse, Doench2014_human, Doench2016, Wang2014, Xiang2021, and Munoz2016—comprising 46,526 unique context sequences. These sequences were characterized by a 20 nucleotide spacer sequence flanked by four nucleotides upstream and a PAM sequence plus three nucleotide contexts downstream, with 45% of sequences incorporating the Chen tracrRNA variant. Each sequence was one-hot encoded to capture the nucleotide arrangement intricately. For the purposes of model training and validation, we adhered to a 5-fold cross-validation procedure, meticulously applied to both training and test sets. Each fold was trained for 150 epochs of training, and evaluated using Spearman's correlation for regression enzymatic activity based on a sequence.

A.3. Protein Tasks

Model Configuration: We use the same architecture for both the protein fitness datasets as well as the TAPE evaluations.

A.3.1. PROTEIN FITNESS PREDICTION TASKS

Experiment Details: For the protein fitness prediction tasks, the Janus was trained across three fitness prediction datasets GB1, Gifford, and GFP. Each dataset contained amino acid sequences of the same length which were one-hot-encoded,

Table 11. Janus Model Configuration for all protein tasks

PARAMETER	55,169
D_MODEL	64
N_LAYERS	1
DROPOUT	0.2
D_INPUT	20
D_OUTPUT	1
PRENORM	TRUE
PGC BLOCK 1	16 HIDDEN DIM, 0.2 DROPOUT
PGC BLOCK 2	128 HIDDEN DIM, 0.2 DROPOUT

input dimension of 20, with the stability and affinity, enrichment, or fluorescence respectively values serving as regression labels . The training was performed for 500 epochs, utilizing the AdamW optimizer with a learning rate of 0.001 and a weight decay of 0.01. The evaluation metric was Spearman's rank correlation coefficient on the validation set, and Mean Squared Error Loss (MSELoss) was used as the loss function.

A.3.2. TAPE EVALUATIONS

Experimental Details: Our evaluation of Janus on TAPE spanned three distinct datasets, addressing fluorescence prediction based on sequence mutations, top-1 accuracy for remote homology detection within super-families, and predictions of structural stability. We adhered to a one-hot encoding scheme for all sequences. For the fluorescence and structural stability tasks, models were trained and subsequently evaluated based on their Spearman regression performance against the training set. Both regression tasks utilized Mean Squared Error (MSE) as the loss criterion, with the AdamW optimizer set to a learning rate of 0.001 and a weight decay of 0.01. The remote homology task, classified as a 7-way classification challenge, followed the same training regime of 500 epochs evaluated by top-1 accuracy on the testset. Here, cross entropy loss was employed, factoring in class sample distributions to inform the loss function, and the same AdamW optimizer settings were maintained.

A.4. Wall-Clock Time Experiments:

In order to investigate the inference speed of different protein language models, we conducted a series of wall-clock time experiments. This study compared our model, Janus, against the protein language models evaluated in the TAPE tasks: ESM1b, DistilProtBert, and TAPE-BERT. All experiments were conducted on an NVIDIA A100 SXM5 GPU with 80GB VRAM to ensure consistent hardware performance across all models. To optimize performance, all transformer-based models (ESM1b, DistilProtBert, and TAPE-BERT) utilized flash-attention. The experiments tested sequences of varying lengths: 64, 128, 256, 512, 1024, 4096, and 16384 amino acids. For each model and sequence length, we measured the time taken for a single forward pass (inference). To ensure accurate timing, we performed a warm-up run before the timed inference for each combination of model and sequence length. Table 12 presents the speed-up of Janus compared to the other models for various sequence lengths, and the Figure below plots the performance in seconds against sequence length (both axes on log scales). The values in the table represent how many times faster Janus performed compared to each respective model.

Table 12. Speed-up of Janus compared to other models

Sequence Length	Janus vs ESM1b	Janus vs DistilProtBert	Janus vs TAPE-BERT
64	10.61	5.29	2.28
128	18.23	7.23	3.71
256	27.24	7.23	4.66
512	49.84	7.12	–
1024	–	7.20	–
4096	–	2.32	–
16384	–	3.16	–

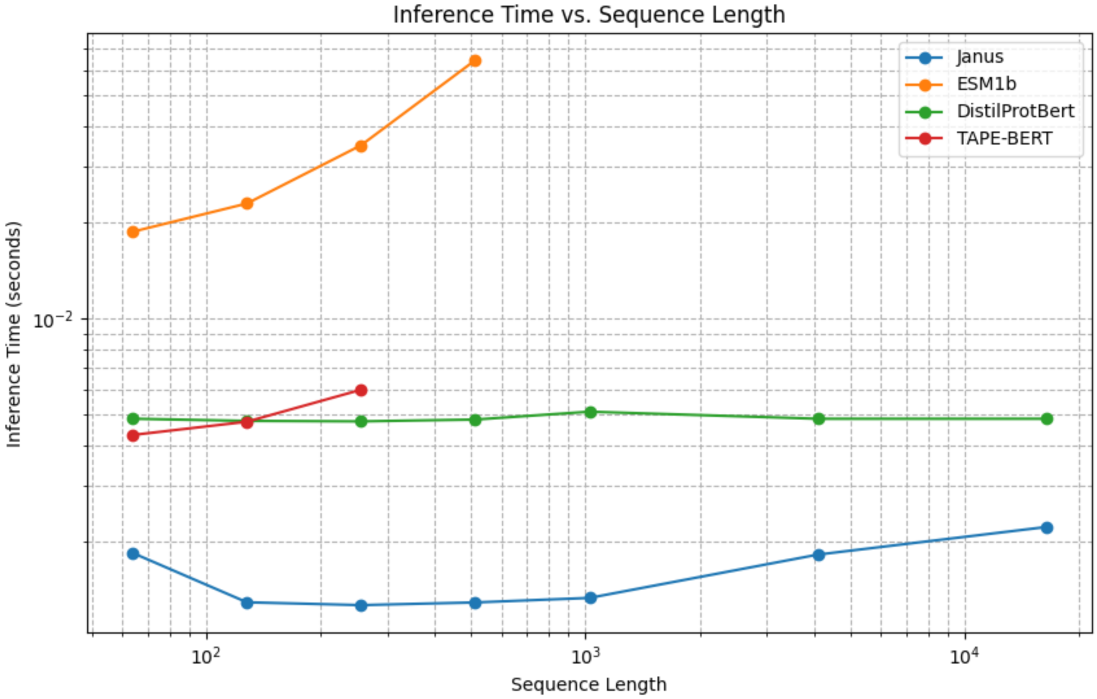

Figure 2. Wall-Clock Time (seconds) evaluation of Janus, ESM1b, DistilProtBERT, and TAPE-BERT on inference against sequence length.

B. Ablation Studies

We present a preliminary investigation of model substitutions in proteomic tasks and intend on extending this investigation to genomic tasks.

B.1. Investigation of Hyena vs BaseConv on ReLSO tasks

In order to discern the impact of the projected gated convolution (PGC) backbone within our model, we conducted a series of ablation studies on the Protein fitness landscape tasks, adhering to the training regimen delineated in Appendix A. These studies were designed to evaluate the effect of substituting the PGC with a Hyena layer and to assess the implications of omitting the backbone entirely to test the S4D component in isolation. Our findings revealed that while replacing the PGC with a Hyena layer did result in a decline in performance, the removal of the backbone to evaluate the S4D alone demonstrated a more pronounced drop across all tasks. This suggests the critical role of the PGC backbone in our model's architecture for maintaining superior performance in protein fitness landscape tasks.

Table 13. Spearman correlation scores for different models on protein fitness datasets for antibody binding (Gifford dataset), antibody fitness (GB1 dataset), and green fluorescent protein (GFP) brightness

Model	Gifford (Ab Binding)	GB1 (Ab Fitness)	GFP
Janus (ours)	**0.50**	**0.61**	**0.86**
Hyena + S4D	0.48	0.60	0.85
S4D	0.48	0.57	0.85