BaseMirror: Leveraging DNA Symmetry for Inference-Time Context Expansion

Anonymous Author(s)
Affiliation
Address
email

Abstract

Genome language models (GLMs) have demonstrated exceptional capabilities in DNA sequence generation and understanding, yet their context-dependent performance is limited by the fixed length of input sequences. To address this limitation, we propose BaseMirror, an inference-time strategy that leverages the symmetry of DNA's double strand to expand the effective context. Our method autoregressively generates tokens along the reverse direction of the reverse-complement strand of a given DNA sequence, then obtains and prepends their complementary bases to the original strand, thereby enriching the model's effective receptive field. We demonstrate that BaseMirror consistently improves generative and discriminative tasks' performance on the GENERator and Evo2 families. For next-base prediction, progressively extending the input sequence leads to consistent performance gains across various input lengths, model sizes, and sampling strategies, with accuracy improvements of up to 4.6%, compared to the original non-extended input. For variant effect prediction on BRCA1, BaseMirror enhances the AUROC for zero-shot classification by up to 5.2%. Moreover, we uncover a scaling phenomenon in which performance increases monotonically with the length of the extended context. Our results highlight the effectiveness of BaseMirror as a lightweight, robust, and scalable solution at inference time through API-based GLM generation.

1 Introduction

Genome language models (GLMs), pre-trained on massive nucleotide corpora and comprising billions of parameters, have demonstrated significant capabilities in modeling DNA sequences, excelling in both generative and discriminative tasks [6, 16, 49, 7, 44]. These autoregressive models generate sequences by predicting each nucleotide based on its preceding context. This context-dependent capability is crucial for producing biologically plausible sequences and modeling complex genomic structures [13]. Notably, recent GLMs like GENERator [44] and Evo2 [7] can now generate biologically meaningful sequences, such as histones, enhancers, and mitochondrial genomes.

However, a fundamental challenge for GLMs is their reliance on the input sequence during inference, which can restrict their effective receptive field, particularly with fixed input lengths. Unlike natural languages, where human-designed prompts or external knowledge can guide model behavior [27, 1], the inherent intricate nature of genomic sequences makes manual prompt engineering or direct information injection largely infeasible [20]. These limitations underscore the need for inference-time strategies that can expand the model's contextual understanding within the autoregressive framework.

In this work, we introduce BaseMirror, a novel inference-time context expansion method. BaseMirror enriches the input sequence by leveraging the inherent reverse-complementary symmetry of DNA. The core principle of our method is rooted in DNA's double-helical structure: the two strands are reverse complements, with adenine (A) pairing with thymine (T) and cytosine (C) with guanine

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

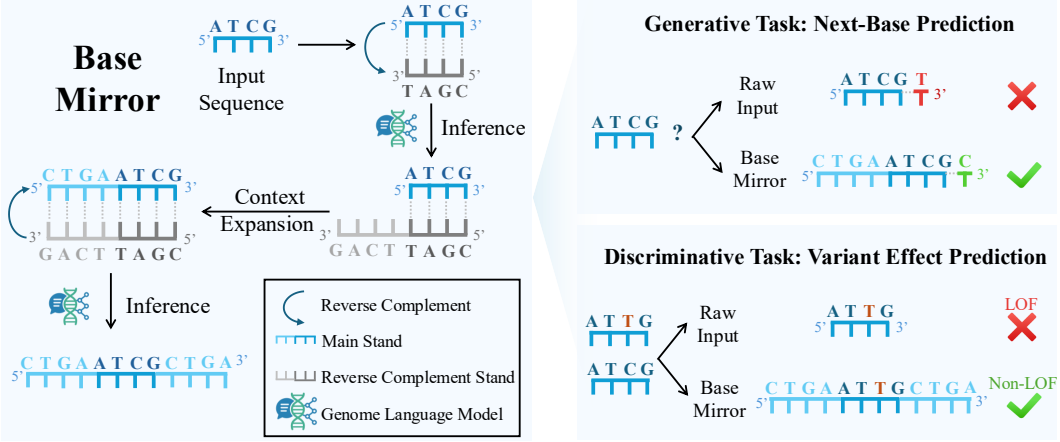

Figure 1: We propose BaseMirror, an inference-time context expansion method that leverages the double-strand symmetry of DNA. Given an input sequence, BaseMirror maps the main DNA strand to its reverse complement and generates the hidden context of the main strand using a genome language model. In this paper, we perform two tasks to demonstrate the effectiveness of BaseMirror. The LOF means that the DNA mutation leads to loss of function, and the Non-LOF means the opposite.

(G) [42]. This complementary pairing ensures that identical genetic information is encoded on both strands, allowing biological mechanisms to recognize patterns on either [33, 28, 47]. We term the operation of mapping a sequence to its reverse complement as *mirroring*.

BaseMirror iteratively leverages this symmetry as illustrated in Figure 1. First, the input sequence is mirrored to its reverse-complement strand, and then the GLM generates tokens along this new strand. These generated bases are subsequently mirrored back to the original strand and prepended to the initial input, effectively serving as expanded context. This process is bidirectional, allowing generation to proceed along the main chain with an enriched contextual view. By augmenting the sequence in this biologically-grounded manner, BaseMirror provides the model with a richer, more informative context without altering model weights and access to task-dependent annotations.

To demonstrate the benefits of expanded context, we conduct experiments on both generative and discriminative tasks. For next k-base prediction, BaseMirror consistently improves accuracy for both Evo2 [7] and GENERator [44] model families, with relative gains up to 4.6%. It also enhances zero-shot classification performance in variant effect prediction (VEP) on BRCA1, a clinical breast cancer dataset [17], with relative gains up to 5.2%. We also demonstrate the generality of BaseMirror on the ClinVar [22], a much larger VEP dataset. These improvements are robust across diverse sequence lengths, numbers of bases to predict, model architectures, and sampling strategies.

Furthermore, our analyses reveal an intriguing inference-time scaling phenomenon facilitated by BaseMirror: downstream performance systematically improves with increased computation at test time, achieved by using longer context expansions. This strong trend is observed in both next-base prediction and variant effect prediction tasks. While the absolute improvements from longer expansions can be moderate, the consistent positive trend highlights BaseMirror's potential as a mechanism to unlock further capabilities of GLMs by investing more computation at inference.

In summary, our contributions are:

- We propose BaseMirror, a novel, model-agnostic inference-time context expansion technique for GLMs that leverages DNA's reverse-complementary structure. It is lightweight and requires only logit-level generation API access without the need of model tuning.
- We demonstrate that BaseMirror significantly improves performance on key genomic tasks, including next-base prediction and zero-shot variant effect prediction Such improvement is robust across sequence lengths, model architectures, and sampling strategies.
- We identify a general inference-time scaling phenomenon: performance on downstream tasks improves with longer context extensions generated by BaseMirror, offering a new avenue for trading test-time computation for accuracy in genomic applications.

2 Related Work

2.1 DNA Reverse Complement in Machine Learning

A fundamental property of DNA is its double-helical structure, where the two strands are reverse complements (RC) of each other (A pairs with T, C pairs with G) [42]. This inherent symmetry implies that both strands carry equivalent genetic information, and many biological processes recognize sequence patterns irrespective of strand orientation [33, 28, 47]. However, standard machine learning architectures, such as conventional Convolutional Neural Networks (CNNs) and Transformers, do not inherently account for RC symmetry [48, 21, 31, 24, 12]. Early approaches typically employed data augmentation by including RC sequences during training [10]. More sophisticated methods embed RC symmetry directly into the model architecture [46]. For example, RC Parameter Sharing (RCPS) in CNNs uses shared weights for filters recognizing forward and RC patterns [34, 3, 8], though such approaches can face optimization difficulties [46]. Recent efforts aim for true RC equivariance, where the model's output transforms predictably when the input is reverse-complemented [26]. Caduceus, for instance, built on the Mamba architecture, introduces specialized modules to process both forward and RC sequences using shared parameters [33]. In contrast, our BaseMirror method is a training-free approach applied, constructing an expanded context at inference time to enhance generation from existing genome language models without altering their architecture or requiring retraining.

2.2 Data Augmentation for DNA Modeling

Data augmentation has proven to be a powerful technique in computer vision (CV) and natural language processing, where it helps improve model generalization, mitigate overfitting, and enhance robustness to distributional shifts [2, 40]. In CV, augmentations like flipping, cropping, and color jitter introduce invariance [45]; in NLP, strategies such as back-translation and synonym substitution introduce semantic diversity without altering meaning [4]. In genomic modeling, however, only a limited number of augmentation strategies have been tailored to the properties of DNA sequences. Reverse complementation is used in training by leveraging the bidirectional nature of DNA strands to double the training data without introducing noise [10]. Another method is genomic shifting, which offsets the input window across the genome to introduce positional variation [14, 36]. More recently, evolution-inspired augmentations such as random point mutations, inversions, and deletions have been proposed to simulate sequence diversity [23]. However, it disregards the underlying functional constraints of biological sequences, potentially introducing unrealistic or non-functional variants. Another method [15] proposes phylogenetic augmentation, using homologous sequences from other species as a data augmentation strategy to improve supervised deep learning models for functional genomics. While it shows promising performance, this method introduces a dependency on external data sources. In this work, our BaseMirror adopts the reverse complement at test time as its core augmentation strategy, requiring no downstream labels, additional sequences, or model fine-tuning.

2.3 In-context Learning in Biology

In-context learning (ICL) is a prominent capability of large language models (LLMs), allowing them to adapt to new tasks based on examples or instructions embedded within the input prompt [27, 1]. This can be achieved with a few examples (few-shot learning) [9, 37], or even without any explicit examples (zero-shot learning) [39]. Recently, the ICL paradigm has begun to gain attention in bioinformatics applications [29, 25, 18]. For example, few-shot ICL has been applied to protein characterization tasks, where general-purpose LLMs have demonstrated performance comparable to, or even exceeding, that of specialized models trained on extensive biological datasets [18]. Zero-shot prediction, where the model leverages its pre-trained knowledge directly with the realistic input sequence, has also shown particular promise for predicting the functional impact of genetic variants [7, 5, 29]. The success of such approaches hinges on the ability of models pre-trained on large-scale genomic sequences to implicitly capture signals of biological function and fitness [6]. Besides, recent studies find that existing ICL methods seem not to work effectively and universally [20, 6]. A key aspect of ICL in genomic contexts is prompt design: most methods rely on carefully crafted prompts that include natural language instructions and input-output example pairs to steer the model's behavior toward the desired task [19, 41]. In contrast, our BaseMirror proposes an inference-time manner to leverage the hidden knowledge of genome language models without any manual design, bypassing the need for natural language instruction or explicit task examples.

3 Method

In this section, we describe our method, BaseMirror, which builds upon the causal framework of a genome language model (GLM). We focus on the importance of conditioning and discuss how BaseMirror overcomes inherent limitations imposed by the context length. In Section 3.1, we introduce the causal modeling of DNA generation by GLMs and propose the context limitation. In Section 3.2, we introduce how our BaseMirror overcomes the inherent context limitation. Finally, we depict the application of BaseMirror in both generative and discriminative tasks in Appendix A.2.

3.1 Autoregressive Model for Genome Sequences

An autoregressive model for sequence prediction generates the next element in the sequence based on the previous elements. In the context of genome sequence generation, let $x_1, x_2, \ldots, x_t$ denote the nucleotide sequence up to position t, where each nucleotide token x_i can be a nucleotide base (*e.g.*, A, T, C, G) or k-mer (*e.g.*, ATGTGG for 6-mer). The nucleotide sequence is described from $5'$ to $3'$ ends by default. The $5'$ and $3'$ ends of a DNA strand refer to the two distinct termini characterized by their chemical groups. These ends are critical for DNA replication and transcription, where new nucleotides are added to the $3'$ end, extending the chain in the $5'$ to $3'$ direction.

For autoregressive generation, the model predicts the next nucleotide token x_{t+1} conditioned on the sequence $x_1, x_2, \ldots, x_t$, formally expressed in Equation (1).

$$P(x_{t+1} \mid x_1, x_2, \ldots, x_t) = \text{softmax}(f(x_1, x_2, \ldots, x_t)) \tag{1}$$

Here, $f()$ is a function, typically implemented by a genome language model, that maps the sequence $x_1, x_2, \ldots, x_t$ to logits over the next possible nucleotide token. The softmax function normalizes these outputs into valid probabilities. This process continues iteratively to generate the full sequence.

Importantly, the prediction of nucleotide token x_{t+1} relies on the preceding sequence context $x_1, x_2, \ldots, x_t$. The quality of this context is crucial for generating plausible sequences [43, 9, 38, 11]. However, the utility of the context is constrained by its length: for generating x_{t+1}, the context is limited to the sequence $x_1, x_2, \ldots, x_t$, hindering the model's ability to capture long-range dependencies.

3.2 Reverse Complement and Context Expansion

We propose that the inherent symmetry of DNA, with its double-stranded structure, may help overcome the context length limitation. In addition to the main strand of DNA, there exists a complementary reverse strand, oriented in the opposite direction ($5'$ to $3'$ opposite to $3'$ to $5'$) of the main strand. This complementary and reverse structure offers an opportunity to extend the context for sequence generation by utilizing the reverse complement strand.

Let $x_1, x_2, \ldots, x_t$ represent the nucleotide sequence of the main strand, from $5'$ to $3'$. The corresponding reverse complement sequence is denoted as $\hat{x}_t, \hat{x}_{t-1}, \ldots, \hat{x}_1$. For example, if x_i is the nucleotide at position i in the main strand, then $\hat{x}_{t-i}$ is the complementary base in the reverse strand at the corresponding position, as shown in Table 1 (contents in black color).

Table 1: DNA double strands described in Markov chain.

	$5'-$	C	T	A	T	G		T	G	G	$-3'$
Main Stand		x_{-2}	x_{-1}	x_1	x_2	x_3	$\cdots$	x_{t-2}	x_{t-1}	x_t	
		|	|	|		|	|	|			
Reverse Complement Stand		$\hat{x}_{t+2}$	$\hat{x}_{t+1}$	$\hat{x}_t$	$\hat{x}_{t-1}$	$\hat{x}_{t-2}$	$\cdots$	$\hat{x}_3$	$\hat{x}_2$	$\hat{x}_1$	
	$3'-$	G	A	T	A	C		A	C	C	$-5'$

By transforming to the reverse complement, we enable predictions for the downstream sequence ($3'$) on the reverse complement strand, which corresponds to the upstream ($5'$) sequence on the main strand. Specifically, the prediction on the reverse complement is $P(\hat{x}_{t+1} \mid x_1, x_2, \ldots, x_t)$. This allows us to predict the *nonexistent* context on the main strand shown in the contents in blue of Table 1. We assume that the token before x_1 is x_{-1}, indicating that x_0 does not exist.

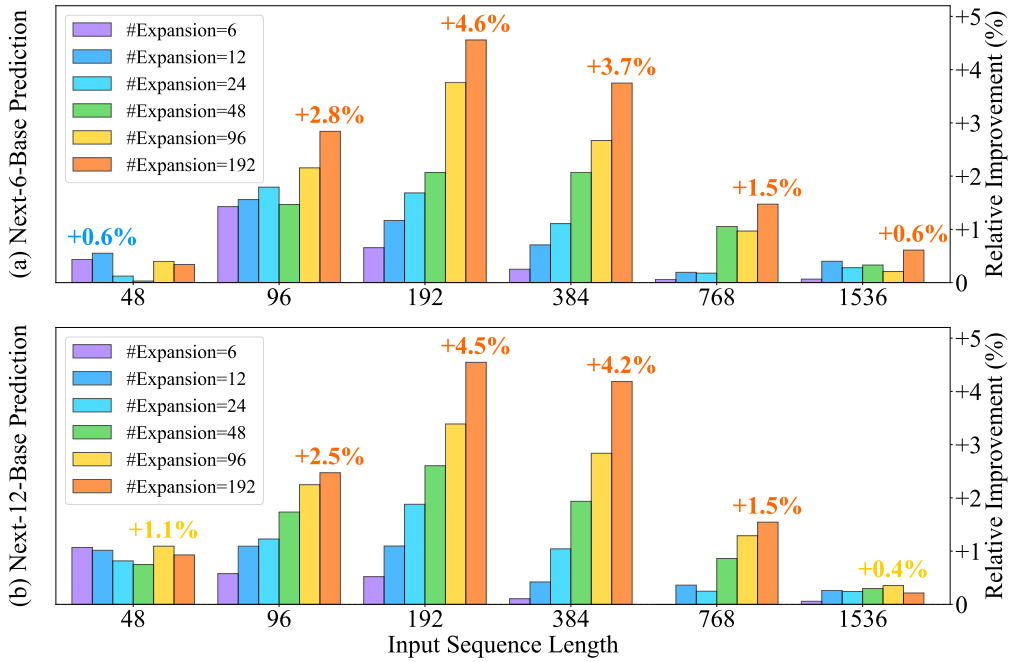

Figure 2: BaseMirror consistently enhances next-base prediction performance across varying input sequence lengths. As the input length increases, the relative accuracy gain over the baseline (original input) also grows. Panels (a) / (b) show the relative improvement results using the GENERator 3B model for predicting the next 6 / 12 bases given the input sequence, respectively.

After generating the downstream of the reverse complement strand, we can map the generated bases to the main strand. Such new generated part is actually the upstream of the main strand, *i.e.*, the context. Detailed steps are shown in the Appendix A.1. We formalize such a process in Equation (2).

$$\begin{aligned} P(x_{-1} \mid x_1, x_2, \ldots, x_t) &= P(\hat{x}_{t+1} \mid x_1, x_2, \ldots, x_t) \\ &= P(\hat{x}_{t+1} \mid \hat{x}_t, \hat{x}_{t-1}, \ldots, \hat{x}_1) \\ &= \text{softmax}(f(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_t)) \end{aligned} \tag{2}$$

The function f here is the same as in Equation (1), as genome language models are pre-trained on both strands [7, 33, 44]. By incorporating the reverse complement strand, we effectively extend the context window for sequence generation. Let N denote the number of tokens in the context expansion, and we can now formalize nucleotide sequence generation as Equation (3).

$$P(x_{t+1} \mid x_{-N}, x_{-(N-1)}, \ldots, x_{-1}, x_1, x_2, \ldots, x_t) \tag{3}$$

According to the introduction above, we can now leverage the DNA symmetry to expand the context during inference. Notably, this symmetry is bidirectional, meaning we can start from the reverse complement strand and, by reversing the process, operate again on the main strand.

4 Experiment

In this section, we evaluate the effectiveness of BaseMirror through experiments on two tasks. First, we present the experimental setup in Section 4.1, which includes task definitions, datasets, evaluation, baselines, models, and hyperparameters. We then provide a detailed analysis of the experiments for both the generative and discriminative tasks in Section 4.2 and Section 4.3, respectively. Finally, we provide an ablation study on generation sampling hyperparameters for the robustness of our method.

4.1 Experimental Settings

We briefly introduce the experimental settings here. The detailed version is shown in Appendix A.3.

Table 2: BaseMirror is generally effective on different models. As the number of expanded bases increases, the performance improvement becomes more pronounced. Besides, a notable scaling effect is observed in model size for both our method and the baselines. The task is predicting the next 6 bases given a sequence of length 192. The red value represents the delta compared to the baseline.

Model	Accuracy (%) by #Expansion					
	0 (Baseline)	6	12	24	48	96
GENERator 1.2B	38.4	38.5+0.1	38.4+0.0	38.7+0.3	39.0+0.6	39.4+1.0
GENERator 3B	40.1	40.3+0.2	40.5+0.4	40.7+0.6	40.9+0.8	41.6+1.5
Evo2 1B base	46.0	46.3+0.3	46.7+0.7	47.0+1.0	47.1+1.1	47.0+1.0
Evo2 7B	52.8	53.1+0.3	53.4+0.6	54.0+1.2	54.2+1.4	54.5+1.7
Evo2 40B	65.4	65.7+0.3	65.8+0.4	66.1+0.7	66.4+1.0	66.6+1.2

Tasks and Dataset The generative task involves predicting the next N bases of a DNA sequence from species like fungi, vertebrate_mammalian, vertebrate_other, invertebrate, protozoa, and plant, similar to next-token prediction [44, 32]. Given a DNA sequence, the model is required to predict the next N bases, with accuracy calculated as $(N_{correct}/N) * 100\%$. For the dataset, we use the released version[1] from [44], filtering sequences to include only the bases A/G/C/T, resulting in 19,941 sequences for next-base prediction. The discriminative task involves predicting the effect of human clinical variants. We adopt the experimental setup from the released version[2] of Evo2 [7], which includes both coding and noncoding regions of the BRCA1 gene [17]. The model is tasked with predicting whether a given variant, represented by the sequence surrounding the SNV variant and its corresponding reference sequence, is pathogenic. All experiments are conducted in a zero-shot setting [7] using GLMs without task-specific tuning, depicted in Appendix A.3.

Models and Hyperparameters We conduct experiments using recently released genome language models from the GENERator [44] and Evo2 [7] families. The Evo2 40B model is accessed through the NVIDIA API[3], while other models are deployed locally (NVIDIA GeForce RTX 4090 GPU, 24G). For the generation process, we employ temperature, top-k, and top-p sampling strategies. In most experiments, we maintain a fixed sampling strategy to limit randomness. Though Evo2 employs single-nucleotide tokenization and its vocabulary contains 512 tokens in total, only four of them correspond to valid DNA bases (A, T, C, and G). Therefore, we set the top-k parameter to 1, effectively disabling sampling. For GENERator, which utilizes 6-mer tokenization, we set the temperature to 1 and top-k to 4 in the majority of our experiments. Additionally, we demonstrate the robustness of BaseMirror across various sampling strategies, as detailed in Table 3.

BaseMirror and Baseline Our method, BaseMirror, is an inference-time approach designed to expand the context of DNA sequences without modifying the underlying pipeline for either the generative or discriminative tasks. Specifically, we expand the task input sequence by a set number of additional bases, referred to as #Expansion. Detailed application of BaseMirror in tasks can be found in Appendix A.2. To demonstrate the effectiveness of the expansion, we compare the performance of our expanded sequences with one using the original input sequence. The baseline corresponds to the raw input sequence with no context expansion, *i.e.*, #Expansion = 0. For consistency, we use the same GLM for context expansion of BaseMirror, and the latter detailed task.

4.2 Generation: Next-base Precision

We conduct experiments across a wide range of input sequence lengths for the next-base prediction task. In the following experiments, we will demonstrate the general effectiveness of our BaseMirror across various (1) input lengths, (2) genome language models, and (3) sampling strategies.

[1] https://huggingface.co/datasets/GenerTeam/next-kmer-prediction

[2] https://github.com/ArcInstitute/evo2/blob/main/notebooks/brca1/brca1_zero_shot_vep.ipynb

[3] https://build.nvidia.com/arc/evo2-40b

BaseMirror consistently improves accuracy across different context lengths. As illustrated in Figure 2, BaseMirror performs well across a range of input sequence lengths and target sequence lengths. Given the substantial variability in absolute accuracy with respect to input sequence length, we present the accuracy improvement, defined as $\frac{ACC_{\#Expansion} - ACC_{baseline}}{ACC_{baseline}} \times 100\%$, to facilitate a direct comparison across different context lengths. The baseline values in Figure 2 corresponds to the original input sequence, with $ACC_{baseline}$ values of 32.8%, 35.6%, 40.1%, 45.7%, 52.3%, and 56.6% for increasing #Expansion values, respectively. Notably, the greatest improvements are observed for mid-range input sequence lengths. This result aligns with expectations, as shorter input sequences may lack sufficient information to generate accurate context during inference.

BaseMirror demonstrates broad efficacy across diverse model families. We evaluate BaseMirror on both the Evo2 and GENERator families, tasked with predicting the next 6 bases from a 192-base sequence, as summarized in Table 2. For both families, sequences utilizing our expanded contexts consistently outperform their corresponding baselines, defined as raw DNA input. Notably, BaseMirror 's effectiveness is evident across a range of model sizes, from 1.2B to 40B parameters, highlighting its robustness and lack of dependence on a specific model capacity.

BaseMirror is robust across diverse sampling hyperparameters. We perform a comprehensive evaluation across various sampling hyperparameters, shown in Table 3. Such a sampling process influences both the context expansion of BaseMirror and the next-base prediction task. Our method consistently performs well across a range of temperatures, top-k, and top-p values. Here, the symbol "/" denotes a neutral setting, *i.e.*, no restriction in the sampling process (top-$k = 0$ and top-$p = 1$). As top-k and top-p increase, the sampling flexibility increases, followed by a notable drop in next-base prediction accuracy. However, BaseMirror can still steadily work on such variable sampling settings, and effectively facilitate the generation task.

Table 3: BaseMirror is robust under diverse generation sampling hyperparameters. The task is predicting the next 6 bases given a DNA sequence of length 384 using GENERator-3B. The top is the default setting in our experiments. The red number is the improvement compared with the baseline.

Temp	Top-k	Top-p	Accuracy (%)	
			Baseline	BaseMirror
1.0	4	1.0	45.7	47.5+1.8
1.2	/	1.00	37.4	38.9+1.5
1.0	/	1.00	39.0	40.5+1.5
0.8	/	1.00	40.6	42.3+1.7
0.6	/	1.00	42.6	44.2+1.6
1.0	**100**	/	42.7	44.1+1.4
1.0	**50**	/	43.4	44.9+1.5
1.0	**10**	/	45.1	46.2+1.1
1.0	**1**	/	46.4	48.2+1.8
1.0	/	**1.00**	39.0	40.5+1.5
1.0	/	**0.99**	39.2	40.7+1.5
1.0	/	**0.90**	40.4	41.5+1.1
1.0	/	**0.70**	41.5	43.2+1.7

A scaling phenomenon emerges across most context lengths and model sizes: as the number of expanded bases increases, performance improves. We hypothesize that BaseMirror creatively exploits an inference-time scaling property of genome language models (GLMs) during the generation process. Furthermore, this paradigm operates at test-time, relying solely on input sequences from downstream tasks. This universal inference-time scaling law is akin to those observed in large language models [30, 35]. We also showcase that BaseMirror generates relatively meaningful context in Figure 6.

4.3 Discrimination: Zero-shot Classification

We experiment on the prediction of the pathogenicity of BRCA1 variants, a binary classification task illustrated in Figure 4. We define the expansion of the original $5' - sequence \rightarrow 3'$ as $3'$ expansion, and the reverse complement of $5' \leftarrow sequence - 3'$ as $5'$ expansion, details of which are shown in Figure 7. We expand the reference sequence using BaseMirror and copy the expanded context to the variant sequence. Notably, expanding the reference and variant sequences independently can lead to inconsistencies, as discussed in Appendix A.7. For our experiments, we perform a log-scale grid search on the number of expanded bases, denoted as #Expansion, for both the $5'$ and $3'$ directions. As shown in Figure 3, we report the relative improvement in AUROC, defined as $\frac{AUROC(x,y) - AUROC(0,0)}{AUROC(0,0)} \times 100\%$, relative to the baseline, *i.e.*, the original input sequence at (0, 0).

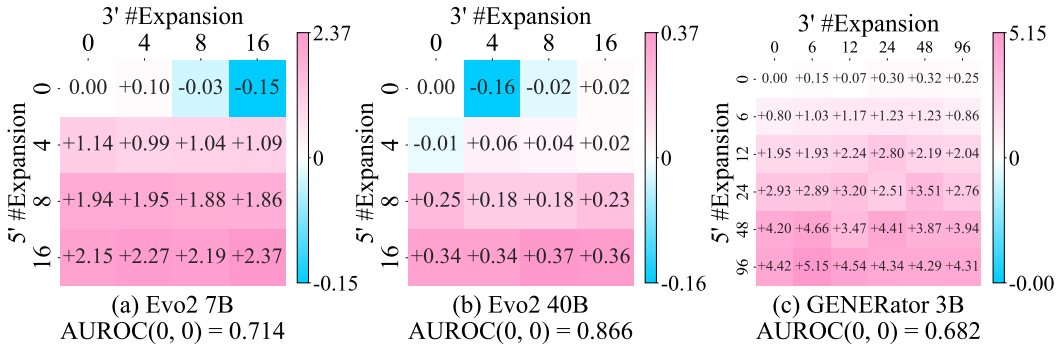

Figure 3: BaseMirror improves BRCA1 classification performance. The relative improvement (%) of AUROC is reported based on the the number of expanded context (log scale) compared to the original input. The blue indicates lower than the baseline (0, 0), while red represents higher.

BaseMirror shows an effective improvement in BRCA1 classification on various genome language models (BLMs). Figure 3 illustrates the relative improvement in AUROC as we iteratively expand the input context during inference, with the x-axis indicating the $3'$ end context expansion and the y-axis denoting the $5'$ end expansion. Across three different models—Evo2 7B, Evo2 40B, and GENERator 3B—we observe consistent improvements in AUROC up to 5.15%, particularly with larger context expansions. This supports the idea that BaseMirror 's context expansion provides a scalable and efficient approach for improving model performance in DNA sequence tasks.

Imbalanced expansion lengths at $3'$ can lead to diminished benefits or even negative effects. In the BRCA1 task, the mutation occurs in the middle of the given sequence, meaning the only difference between the reference and variant sequences is the nucleotide base at the center. When the $3'$ context length is expanded excessively, a hallucination phenomenon arises, where the generated sequence does not contribute effectively to variant significance classification. In contrast, expanding the $5'$ context consistently improves the results. We hypothesize that this difference stems from the zero-shot classification mechanism [7], which captures the influence of upstream mutations on downstream bases. Consequently, a longer $5'$ context, such as $x_{-N}, \ldots, x_{-2}, x_{-1}$, is more beneficial than a longer $3'$ context, such as $x_{t+1}, x_{t+2}, \ldots, x_N$.

BaseMirror also demonstrates strong performance on the ClinVar variant effect prediction dataset[22], extending its effectiveness beyond BRCA1. To assess generalization, we evaluate our approach on the full ClinVar dataset, comprising 40,976 samples[4], using the GENERator-3B model. With a 510-length input expanded by 192 bases, BaseMirror achieve an AUROC of 0.8349, a notable improvement over the baseline (without context expansion) of 0.8224. These results, on a dataset over ten times larger than the BRCA1 set, confirm that BaseMirror consistently enhances performance even when the baseline AUROC is already high.

5 Conclusion

In this paper, we introduce a novel context expansion method, BaseMirror, which leverages the double-strand symmetry of DNA through genome language models (BLMs). By mapping the input DNA sequence to its reverse complement, BaseMirror generates hidden contexts in an iterative manner. Notably, this approach operates purely on the input sequence at inference time, requiring no model parameter tuning, and can be deployed using BLMs' logits API from cloud servers. Our experimental results across generative and discriminative tasks demonstrate the broad applicability of BaseMirror. A key insight is the inference-time scalability: as the context expansion computation increases, we observe a corresponding improvement in task performance. Additionally, one limitation of BaseMirror is that the imbalanced usage of the $3'$ expansion may negatively impact the task.

[4]https://huggingface.co/datasets/songlab/clinvar

References

[1] Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

[2] Khaled Alomar, Halil Ibrahim Aysel, and Xiaohao Cai. Data augmentation in classification and segmentation: A survey and new strategies. *Journal of Imaging*, 9(2):46, 2023.

[3] Jakub M Bartoszewicz, Anja Seidel, Robert Rentzsch, and Bernhard Y Renard. Deepac: predicting pathogenic potential of novel dna with reverse-complement neural networks. *Bioinformatics*, 36(1):81–89, 2020.

[4] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, 2022.

[5] 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.

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

[7] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, et al. Genome modeling and design across all domains of life with evo 2. *bioRxiv*, pages 2025–02, 2025.

[8] Richard C Brown and Gerton Lunter. An equivariant bayesian convolutional network predicts recombination hotspots and accurately resolves binding motifs. *Bioinformatics*, 35(13):2177–2184, 2019.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Zhen Cao and Shihua Zhang. Simple tricks of convolutional neural network architectures improve dna–protein binding prediction. *Bioinformatics*, 35(11):1837–1843, 2019.

[11] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of the royal society interface*, 15(141):20170387, 2018.

[12] Jim Clauwaert and Willem Waegeman. Novel transformer networks for improved sequence labeling in genomics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(1):97–106, 2020.

[13] Micaela E Consens, Cameron Dufault, Michael Wainberg, Duncan Forster, Mehran Karimzadeh, Hani Goodarzi, Fabian J Theis, Alan Moses, and Bo Wang. Transformers and genome language models. *Nature Machine Intelligence*, pages 1–17, 2025.

[14] Bernardo P de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark. Deepstarr predicts enhancer activity from dna sequence and enables the de novo design of synthetic enhancers. *Nature genetics*, 54(5):613–624, 2022.

[15] Andrew G Duncan, Jennifer A Mitchell, and Alan M Moses. Improving the performance of supervised deep learning for regulatory genomics using phylogenetic augmentation. *Bioinformatics*, 40(4):btae190, 2024.

[16] Haonan Feng, Lang Wu, Bingxin Zhao, Chad Huff, Jianjun Zhang, Jia Wu, Lifeng Lin, Peng Wei, and Chong Wu. Benchmarking dna foundation models for genomic sequence classification. *bioRxiv*, 2024.

[17] 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.

[18] Sin-Hang Fung, Zhenghao Zhang, Ran Wang, Chen Miao, Brian Shing-Hei Wong, Kelly Yichen Li, Chenyang Hong, Jingying Zhou, Kevin Y Yip, Stephen Kwok-Wing Tsui, et al. Few-shot in-context learning with large language models for antibody characterization. *bioRxiv*, pages 2025–02, 2025.

[19] Jiyue Jiang, Zikang Wang, Yuheng Shan, Heyan Chai, Jiayi Li, Zixian Ma, Xinrui Zhang, and Yu Li. Biological sequence with language model prompting: A survey. *arXiv preprint arXiv:2503.04135*, 2025.

[20] Pranav Kantroo, Günter P Wagner, and Benjamin B Machta. In-context learning can distort the relationship between sequence likelihoods and biological fitness. *arXiv preprint arXiv:2504.17068*, 2025.

[21] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[22] Melissa J Landrum, Jennifer M Lee, George R Riley, Wonhee Jang, Wendy S Rubinstein, Deanna M Church, and Donna R Maglott. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic acids research*, 42(D1):D980–D985, 2014.

[23] Nicholas Keone Lee, Ziqi Tang, Shushan Toneyan, and Peter K Koo. Evoaug: improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations. *Genome Biology*, 24(1):105, 2023.

[24] Jiahao Li, Zhourun Wu, Wenhao Lin, Jiawei Luo, Jun Zhang, Qingcai Chen, and Junjie Chen. ienhancer-elm: improve enhancer identification by extracting position-related multiscale contextual information based on enhancer language models. *Bioinformatics Advances*, 3(1): vbad043, 2023.

[25] Jiajia Liu, Mengyuan Yang, Yankai Yu, Haixia Xu, Kang Li, and Xiaobo Zhou. Large language models in bioinformatics: applications and perspectives. *arXiv preprint arXiv:2401.04155*, 2024.

[26] Vincent Mallet and Jean-Philippe Vert. Reverse-complement equivariant networks for dna sequences. *Advances in neural information processing systems*, 34:13511–13523, 2021.

[27] Haitao Mao, Guangliang Liu, Yao Ma, Rongrong Wang, Kristen Johnson, and Jiliang Tang. A survey to recent progress towards understanding in-context learning. *arXiv preprint arXiv:2402.02212*, 2024.

[28] Guillaume Marçais, C_S Elder, and Carl Kingsford. k-nonical space: sketching with reverse complements. *Bioinformatics*, 40(11):btae629, 2024.

[29] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

[30] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[31] Daniel Quang and Xiaohui Xie. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 166: 40–47, 2019.

[32] Melissa Sanabria, Jonas Hirsch, Pierre M Joubert, and Anna R Poetsch. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8): 911–923, 2024.

[33] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *arXiv preprint arXiv:2403.03234*, 2024.

[34] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Reverse-complement parameter sharing improves deep learning models for genomics. *BioRxiv*, page 103663, 2017.

[35] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[36] Shushan Toneyan, Ziqi Tang, and Peter K Koo. Evaluating deep learning for predicting epigenomic profiles. *Nature machine intelligence*, 4(12):1088–1100, 2022.

[37] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022.

[40] Zaitian Wang, Pengfei Wang, Kunpeng Liu, Pengyang Wang, Yanjie Fu, Chang-Tien Lu, Charu C Aggarwal, Jian Pei, and Yuanchun Zhou. A comprehensive survey on data augmentation. *arXiv preprint arXiv:2405.09591*, 2024.

[41] Zeyuan Wang, Binbin Chen, Keyan Ding, Jiawen Cao, Ming Qin, Yadan Niu, Xiang Zhuang, Xiaotong Li, Kehua Feng, Tong Xu, et al. Multi-purpose controllable protein generation via prompted language models. *bioRxiv*, pages 2024–11, 2024.

[42] James D Watson and Francis HC Crick. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[44] Wei Wu, Qiuyi Li, Mingyang Li, Kun Fu, Fuli Feng, Jieping Ye, Hui Xiong, and Zheng Wang. Generator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

[45] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022.

[46] Hannah Zhou, Avanti Shrikumar, and Anshul Kundaje. Benchmarking reverse-complement strategies for deep learning models in genomics. *bioRxiv*, pages 2020–11, 2020.

[47] Hannah Zhou, Avanti Shrikumar, and Anshul Kundaje. Towards a better understanding of reverse-complement equivariance for deep learning models in genomics. In *Machine Learning in Computational Biology*, pages 1–33. PMLR, 2022.

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

[49] Xiao Zhu, Chenchen Qin, Fang Wang, Fan Yang, Bing He, Yu Zhao, and Jianhua Yao. Cd-gpt: a biological foundation model bridging the gap between molecular sequences through central dogma. *bioRxiv*, pages 2024–06, 2024.

A Technical Appendices and Supplementary Material

A.1 Detailed Process of BaseMirror

We describe the detailed process involved in generating the DNA strands based on the Markov chain model outlined in the main text. This process involves inputting an initial DNA sequence, generating the reverse complement strand, and mapping generated bases to the main strand using defined transition rules. The methodology involves simulating both the main and reverse complement strands as shown in Table 4, Table 5, Table 6, and Table 7.

A.2 Application in Tasks

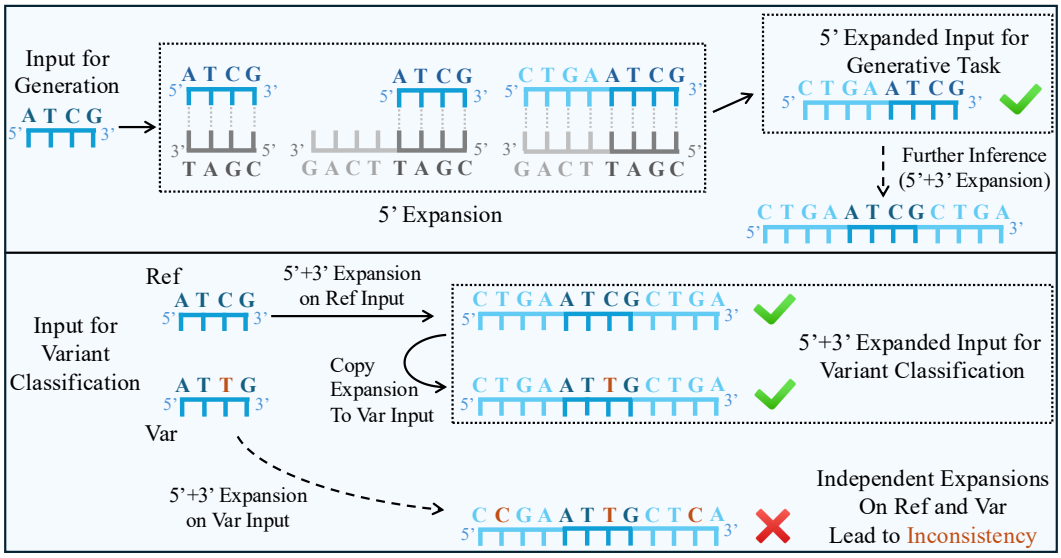

Figure 4: We present an overview of the $5'$ and $3'$ expansion of BaseMirror in downstream applications. The top section illustrates the $5'$ context expansion for the generative task. The lower section outlines the process of variant classification, where both the $5'$ and $3'$ expansions ($5' + 3'$) are applied to the reference sequence (Ref). These expansions are then transferred to the variant sequence (Var).

For detailed application in generation and variant classification tasks, we describe the usage of BaseMirror in Figure 4. And the variant classification mechanism for BRCA1 is described in A.3.3. We define the expansion of the original $5' - sequence \rightarrow 3'$ as $3'$ expansion, and the reverse complement of $5' \leftarrow sequence - 3'$ as $5'$ expansion, details of which are shown in Figure 7. The upper section illustrates the $5'$ context expansion for the generative task. Additionally, further inference can facilitate the formulation of a $5' + 3'$ expansion. The lower section depicts the variant classification process, where both the $5' + 3'$ expansions are applied to the reference sequence. These expansions are then transferred to the variant sequence.

Notably, independent expansions on the reference and variant sequences may result in inconsistencies for variant classification. During the development of our method, we observed a decline in the AROC for zero-shot classification as the expansion length increased. Since the reference sequence serves as a background, such inconsistencies undermine the measurement of the mutation. In Appendix A.7, we quantitatively demonstrate the detrimental effect of these inconsistencies.

Step 1: Input DNA Sequence The input DNA sequence is represented as a series of states in a Markov chain model. The main strand of the DNA sequence is defined in Table 4.

Table 4: Input DNA sequence described in Markov chain.

	$5'-$	A	T	G		T	G	G	$-3'$
Main Stand		x_1	x_2	x_3	$\cdots$	x_{t-2}	x_{t-1}	x_t	

This sequence corresponds to the states $x_1, x_2, x_3, \ldots, x_t$ of the Markov chain. The model processes the sequence by transitioning from one state to the next.

Step 2: DNA Double-Strand Representation Using the initial sequence, the model generates both the main strand and the reverse complement strand. The reverse complement strand is derived by replacing each base in the main strand with its complement, shown in Table 5

Table 5: DNA double strands described in Markov chain.

	$5'-$	A	T	G		T	G	G	$-3'$
Main Stand		x_1	x_2	x_3	$\cdots$	x_{t-2}	x_{t-1}	x_t	
		|	|	|		|	|	|	
Reverse Complement Stand		$\hat{x}_t$	$\hat{x}_{t-1}$	$\hat{x}_{t-2}$	$\cdots$	$\hat{x}_3$	$\hat{x}_2$	$\hat{x}_1$	
	$3'-$	T	A	C		A	C	C	$-5'$

The corresponding Markov chain states for both strands are given by $x_1, x_2, \ldots, x_t$ for the main strand and $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_t$ for the reverse complement strand.

Step 3: Generating Downstream of the Reverse Complement Strand To generate the downstream sequence of the reverse complement strand, the model applies transitions based on previously defined transition probabilities. The sequence downstream from the initial reverse complement strand is generated as shown in Table 6.

Table 6: Generating the downstream of the reverse complement strand.

	$5'-$			A	T	G		T	G	G	$-3'$
Main Stand				x_1	x_2	x_3	$\cdots$	x_{t-2}	x_{t-1}	x_t	
				|	|	|		|	|	|	
Reverse Complement Stand		$\hat{x}_{t+2}$	$\hat{x}_{t+1}$	$\hat{x}_t$	$\hat{x}_{t-1}$	$\hat{x}_{t-2}$	$\cdots$	$\hat{x}_3$	$\hat{x}_2$	$\hat{x}_1$	
	$3'-$	G	A	T	A	C		A	C	C	$-5'$

Step 4: Mapping the Generated Reverse Complement Bases to the Main Strand Finally, the model maps the generated bases of the reverse complement strand back to the corresponding bases in the main strand. This is done using a set of mapping rules derived from the transitions in the Markov model shown in Table 7.

Table 7: Mapping the generated bases of the reverse complement to the main strand.

	$5'-$	C	T	A	T	G		T	G	G	$-3'$
Main Stand		x_{-2}	x_{-1}	x_1	x_2	x_3	$\cdots$	x_{t-2}	x_{t-1}	x_t	
		|	|	|		|	|	|		
Reverse Complement Stand		$\hat{x}_{t+2}$	$\hat{x}_{t+1}$	$\hat{x}_t$	$\hat{x}_{t-1}$	$\hat{x}_{t-2}$	$\cdots$	$\hat{x}_3$	$\hat{x}_2$	$\hat{x}_1$	
	$3'-$	G	A	T	A	C		A	C	C	$-5'$

According to the introduction above, we can now leverage the DNA symmetry to expand the context during inference. Notably, this symmetry is bidirectional, meaning we can start from the reverse complement strand and, by reversing the process, operate again on the main strand. This method allows for the precise simulation and generation of DNA sequences with both forward and reverse complement strands modeled using Markov chains.

A.3 Detailed Experimental Settings

A.3.1 Generative Task: Next k-Base Prediction

The generative task involves predicting the next N bases of a DNA sequence from fungal species. This task is similar to next-token prediction [44, 32], but due to varying tokenization units in genome

language models (GLMs), we modify it to a next-base prediction task for fair comparison. Given a DNA sequence, the model is required to predict the next N bases, with accuracy calculated as $(N_{correct}/N)*100\%$. We set N to 6 or 12, as some models in our experiments use 6-mer tokenization. For the dataset, we use the released version[5] from [44], filtering sequences to include only the bases ATCG, resulting in 19,941 sequences for next-base prediction. It is important to note that the length of the input sequence varies across different experiments. This variability will be clarified independently for each experiment as needed.

A.3.2 Discriminative Task: Variant Effect Prediction

The discriminative task involves predicting the effect of human clinical variants, specifically binary classification for biologically significant sequence variations. The ClinVar [22] dataset comprises 40,976 samples. For BRCA1 [17] dataset, We adopt the experimental setup from the released version[6] of Evo2 [7], which includes both coding and noncoding regions of the BRCA1 gene. The dataset consists of 3,893 pairs of variant and reference sequences, with 3,070 labeled as loss of function (LOF) and 823 as function/intermediate (Non-LOF). The model is tasked with predicting whether a given variant, represented by the sequence surrounding the SNV variant and its corresponding reference sequence, is pathogenic. All experiments are conducted in a zero-shot setting [7] using GLMs without task-specific fine-tuning. Specifically, the GLMs predict the logits for both the mutant and reference sequences. The variant significance is then determined by computing the delta between the predicted log-likelihoods of the mutant and reference sequences. Zero-shot details are shown in Appendix A.3.3. We use a sequence length of 512 and report the AUROC score.

A.3.3 Zero-shot Mechanism of Variant Effect Prediction

In this section, we introduce the detailed mechanism of zero-shot implementation on variant classification. In a zero-shot setting, the model is not explicitly trained on labeled variant effect data but instead leverages its pretrained knowledge of genomic sequences to assess mutation impact directly from sequence likelihoods. The input to the model consists of a pair of sequences: a reference sequence (ref) and a variant sequence (var), differing by a single nucleotide variant (SNV).

Here, the reference sequence $\mathbf{x}^{\text{ref}}$ represents the wild-type (non-mutated) version of a genomic region, while $\mathbf{x}^{\text{var}}$ is an otherwise identical sequence that contains a SNV at a specific position. The task is to predict whether the variant results in loss of function (LoF) or the opposite (Non-LOF). Given a reference sequence $\mathbf{x}^{\text{ref}}$ and a variant sequence $\mathbf{x}^{\text{var}}$, we compute a log-likelihood score for each sequence by averaging the model's log-probabilities over all positions:

$$\log p(\mathbf{x}) = \frac{1}{L} \sum_{t=1}^{L} \log p(x_t \mid \mathbf{x}_{<t}) \tag{4}$$

Here, L is the sequence length, and $p(x_t \mid \mathbf{x}_{<t})$ is the probability assigned to the true nucleotide at position t under the model's autoregressive output. In practice, this is computed by applying a log-softmax over the model's output logits at each position and gathering the value corresponding to the ground-truth token. Then the delta likelihood score between the reference and variant is calculated as:

$$\Delta\mathcal{L} = \log p(\mathbf{x}^{\text{ref}}) - \log p(\mathbf{x}^{\text{var}}) \tag{5}$$

This score serves as a proxy for mutation impact, with higher values indicating greater disruption under the model's learned distribution. To evaluate the classification performance of this approach, the Area Under the Receiver Operating Characteristic (AUROC) is computed based on the delta likelihood score. Note that the AUROC is calculated on $-$score: roc_auc_score(y_true, -brca1_df['evo2_delta_score'])[7].

[5] https://huggingface.co/datasets/GenerTeam/next-kmer-prediction

[6] https://github.com/ArcInstitute/evo2/blob/main/notebooks/brca1/brca1_zero_shot_vep.ipynb

[7] `https://github.com/ArcInstitute/evo2/blob/main/notebooks/brca1/brca1_zero_shot_vep.ipynb`

A.3.4 Models and Hyperparameters

We conduct experiments using recently released genome language models from the GENERator [44] and Evo2 [7] families. Specifically, our experiments involve five models, Evo2 1B base, Evo2 7B, Evo2 40B, GENERator 1.2B, and GENERator 3B, for a general conclusion. The Evo2 40B model is accessed through the NVIDIA API[8], while other models are deployed locally. For the generation process, we employ temperature, top-k, and top-p sampling strategies. The temperature controls the flexibility of the sampling, with higher temperatures promoting greater variability. Top-k restricts the sampling to the k tokens with the highest probabilities, while top-p selects tokens whose cumulative probability is less than or equal to p.

In most experiments, we maintain a fixed sampling strategy to limit randomness. Since Evo2 employs single-nucleotide tokenization and has a vocabulary size of 512, restricting the task to four valid tokens, we set the top-k parameter to 1, effectively disabling sampling. For GENERator, which utilizes 6-mer tokenization, we set the temperature to 1 and top-k to 4 in the majority of our experiments. Additionally, we assess the impact of different sampling hyperparameters, as detailed in Table 3, demonstrating the robustness of BaseMirror across various sampling strategies.

A.3.5 BaseMirror and Baseline

Our method, BaseMirror, is an inference-time approach designed to expand the context of DNA sequences without modifying the underlying pipeline for either the generative or discriminative tasks. Specifically, we expand the task input sequence by a set number of additional bases, referred to as #Expansion. The baseline corresponds to the raw input sequence with no context expansion, *i.e.*, #Expansion = 0. For consistency, we use the same GLM for context expansion of BaseMirror, and the latter detailed task.

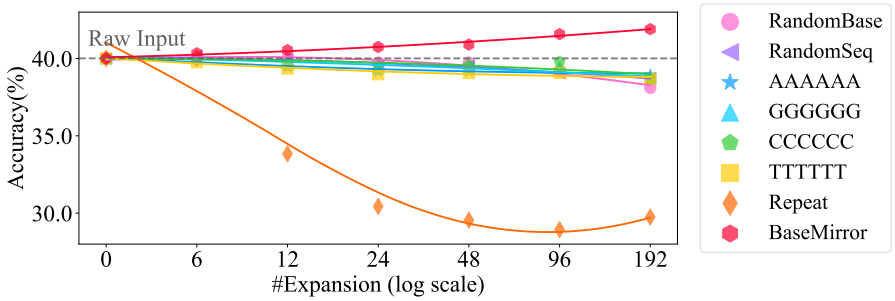

Figure 5: BaseMirror indicates generating positively meaningful context, rather than just random expansion. The task is to predict the next 6 bases given a sequence of length 192. For baselines, the RandomBase represents expanding context randomly, while the RandomSeq means randomly selecting bases from the input sequence as context. The other four use the fixed A/T/C/G sequences. The Repeat mode copies the end of the given sequence to serve as the expanded context.

A.4 Context Expansion Modes

BaseMirror effectively generates positively meaningful context for given input sequences. In Figure 6, we compare our method with different context expansion modes, the Python code of which is shown in the Appendix A.5. We randomly generate #Expansion bases and add such context to the original input, termed as RandomBase. Since the base distribution of sequences might be different, we slightly change the setting by selecting bases from the input sequence randomly, resulting in RandomSeq. Furthermore, we also experiment with fixed sequences such as all A/G/C/T for context expansion. In Figure 5 of Appendix A.4, we also experiment with the "Repeat" mode, which copies the end of the given sequence as the expanded context. BaseMirror is the only method that could keep growing by the number of expanded contexts, demonstrating its generation of meaningful context.

The result of all modes is shown in Figure 5. As the performance of the Repeat mode drops sharply, we only leave the other seven modes in the main text in Figure 6. Notably, the context generated

[8]https://build.nvidia.com/arc/evo2-40b

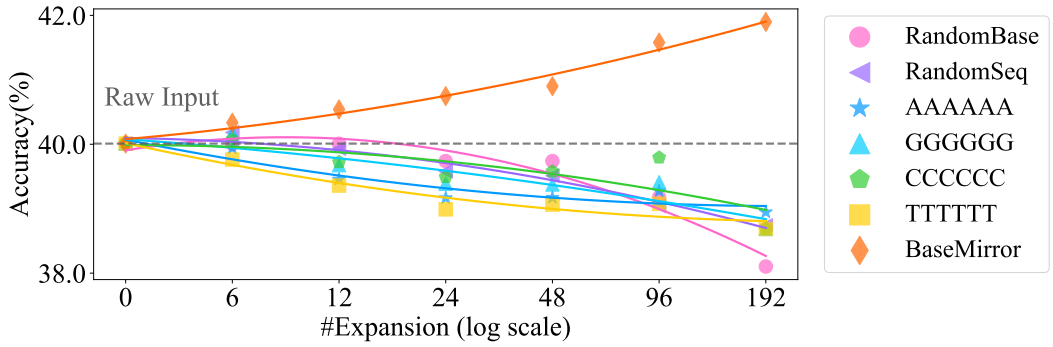

Figure 6: BaseMirror indicates generating positively meaningful context, rather than just random expansion. The task is to predict the next 6 bases given a sequence of length 192. For baselines, the RandomBase represents expanding context randomly, while the RandomSeq means randomly selecting bases from the input sequence as context. The other four use the fixed A/G/C/T sequences.

by BaseMirror can help the further next-base generation, while the copy of a piece of biologically meaningful context fails. Shown in Figure 5, the orange line, "Repeat" mode, copies the end of the given sequence as the expanded context, code of which is shown in the Appendix A.5. Both the RandomSeq and Repeat modes make the prediction accuracy drop sharply as the length of the repeat increases. We reckon that even biologically meaningful contexts can be harmful to the next-base prediction. Instead, BaseMirror leverages genome language models' general knowledge gained from large-scale pre-training and can generate positively meaningful content.

A.5 Python Implementation of Baseline Context Modes

The Python implementation of baseline context modes is shown as follows:

```python
def generate_context(mode:str, sequences: List[str], new_length: int):
    if mode == "RandomSeq":
        expanded_sequences = [
            "".join(random.choice(seq) for _ in range(new_length))
            for seq in sequences
        ]
    elif mode == "RandomBase":
        expanded_sequences = [
            "".join(random.choice('TAGC') for _ in range(new_length))
            for _ in range(len(sequences))
        ]
    elif mode == "Repeat":
        expanded_sequences = []
        for seq in sequences:
            expanded_seq = ""
            while len(expanded_seq) < new_length:
                expanded_seq = expanded_seq + seq[-new_length:]
            expanded_sequences.append(expanded_seq[-new_length:])
    elif mode in ["A", "T", "C", "G"]:
        expanded_sequences = [
            mode * new_length for _ in range(len(sequences))
        ]
    else:
        raise ValueError(f"Invalid mode: {mode}")
    return expanded_sequences
```

A.6 Details of BRCA1 Variant Classification

In this section, we elaborate on some experimental details and results for the BRCA1 variant classification task owing to limited space. As shown in Figure 7, we perform a detailed analysis to assess the impact of varying context lengths on the BRCA1 task. The experiment focuses on the

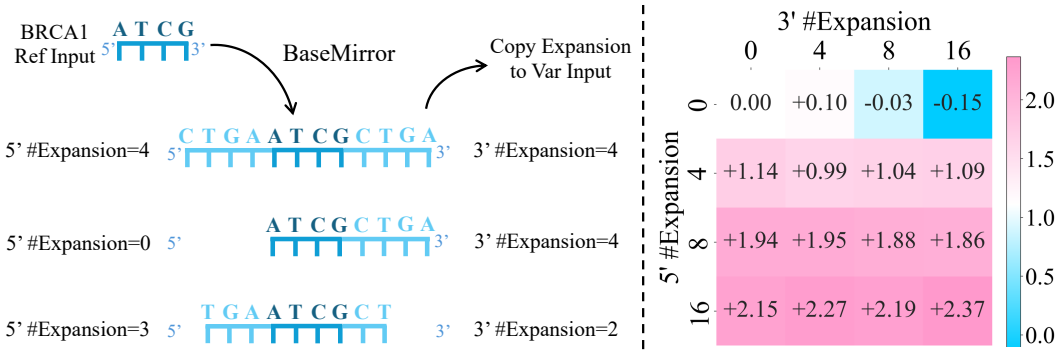

Figure 7: We perform detailed experiments on the context lengths on the BRCA1 task. We design the $5'$ and $3'$ expansion shown on the left. On the right, the relative improvement (%) of AUROC matrix is reported using a log scale, compared with the original input at (0, 0). Experiments are conducted on the Evo2 7B model with a baseline AUROC of 0.714 at (0, 0).

effects of both $5'$ and $3'$ expansions of the reference input sequences. To facilitate our investigation, we vary the context lengths by adjusting the number of bases considered on both ends of the reference sequence.

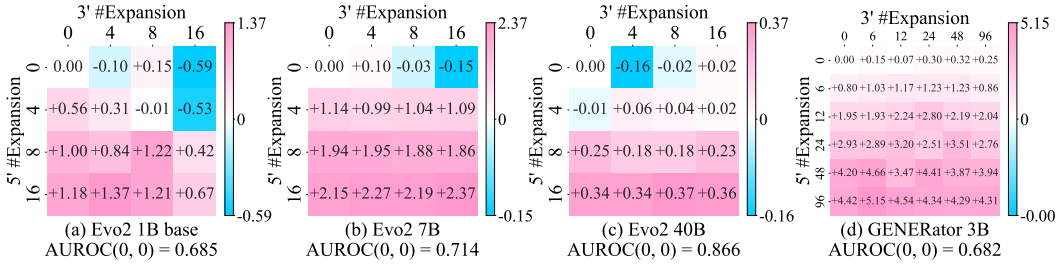

Figure 8: BaseMirror consistently improves on the BRCA1 classification. Relative improvement (%) on AUROC at the original input is reported by the the number of expanded context (log scale).

For each input sequence, we generate multiple expansions by modifying the context around the reference sequence, shown in Figure 8. We test several configurations, where the $5'$ and $3'$ context lengths are expanded in increments of 4 base pairs (with the values ranging from 0 to 16 for both ends). For each configuration, we use a model to compute the area under the receiver operating characteristic curve (AUROC) to evaluate performance. The relative improvements (%) of AUROC values are reported as differences compared to the baseline model performance using the original input (at 0,0 expansion).

These findings indicate that larger context expansions on both ends of the sequence are beneficial for the model's performance on the BRCA1 task, with diminishing returns as the expansion exceeds a certain threshold. The log scale representation of the AUROC values provides a clear visual indication of the improvements achieved with various context lengths, reinforcing the importance of proper sequence context in tasks requiring DNA sequence analysis.

A.7 Consistency in Variant Classification

As mentioned in the lower section of Figure 4, the consistency of context expansion is crucial for multi-input tasks, such as the BRCA1 variant classification, since the reference sequence serves as the background to compare and measure the significance of the mutation in the variant sequence. Once both the reference sequence and the variant sequence expand the context independently, the resulting expanded sequences will have more differences than the original base mutation. The influence of context expansion inconsistency is shown in Figure 9.

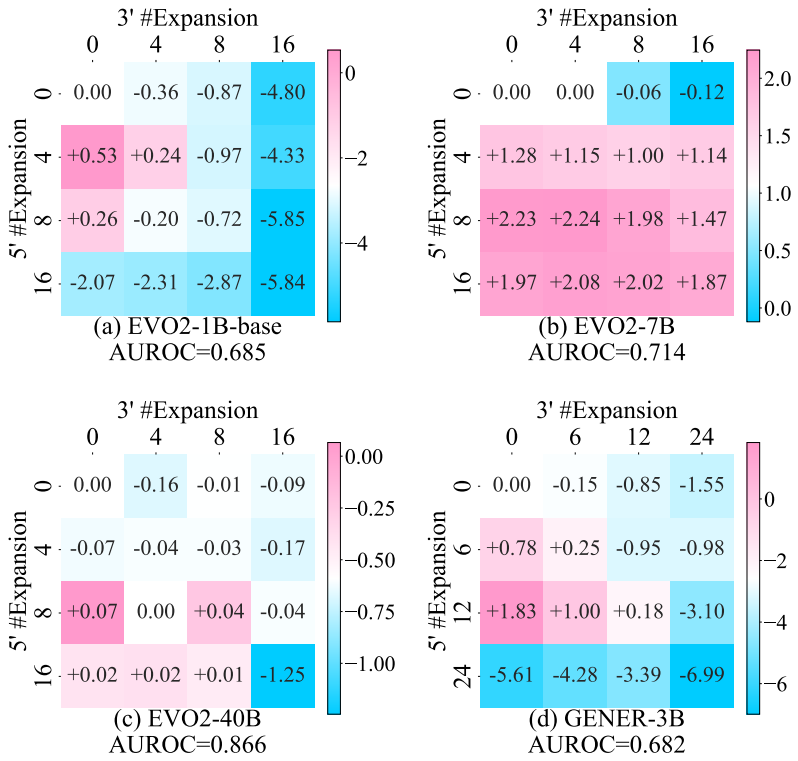

Figure 9: Independent context expansions of the reference sequence and the variant sequence lead to failures. Relative improvement (%) on AUROC at the original input is reported by the the number of expanded context (log scale).