Language Models for Controllable DNA Sequence Design

Anonymous authors
Paper under double-blind review

Abstract

We consider controllable DNA sequence design, where sequences are generated by conditioning on specific biological properties. While language models (LMs) such as GPT and BERT have achieved remarkable success in natural language generation, their application to DNA sequence generation remains largely underexplored. In this work, we introduce ATGC-Gen, an Automated Transformer Generator for Controllable Generation, which leverages cross-modal encoding to integrate diverse biological signals. ATGC-Gen is instantiated with both decoder-only and encoder-only transformer architectures, allowing flexible training and generation under either autoregressive or masked recovery objectives. We evaluate ATGC-Gen on representative tasks including promoter and enhancer sequence design, and further introduce a new dataset based on ChIP-Seq experiments for modeling protein binding specificity. Our experiments demonstrate that ATGC-Gen can generate fluent, diverse, and biologically relevant sequences aligned with the desired properties. Compared to prior methods, our model achieves notable improvements in controllability and functional relevance, highlighting the potential of language models in advancing programmable genomic design.

1 Introduction

DNA sequence design is a transformative scientific endeavor that revolutionizes our understanding of biology and catalyzes major advances across healthcare, agriculture, and environmental conservation, etc (Zrimec et al., 2022; Killoran et al., 2017). In these tasks, it is commonly expected that the generated DNA sequence achieves some biological outcomes (Uehara et al., 2025; Li et al., 2025). Thus, controllable generation is of practical significance (Nguyen et al., 2024a) in which the sequence generation is guided by various properties, *e.g.,* binding to specific proteins or exhibiting particular transcription activation likelihoods.

Recent advances in generative models have triggered widespread interests in DNA sequence design using diffusion (Avdeyev et al., 2023; Li et al., 2024; Sarkar et al., 2024) and flow matching (Stark et al., 2024) generative methods. These methods have shown promise, particularly in modeling global structure and optimizing over continuous latent spaces. However, they are not inherently tailored to discrete, symbolic sequence generation—properties that are central to DNA sequences. While those methods are in general effective in generation task, they are not naturally designed for generating discrete and variable-length sequences. In contrast, Language models (LMs) offer an alternative perspective: they are naturally suited for discrete, variable-length generation and have achieved remarkable success in analogous domains like natural language. In this work, we explore the potential of LMs for controllable DNA sequence design, proposing them as a complementary approach to existing generative methods.

In this paper, we aim to explore the use of transformer-based language models for DNA sequence design conditioned on specific biological properties (Vaswani et al., 2017; Stark et al., 2024; Sarkar et al., 2024; Su et al., 2025). To this end, we develop ATGC-Gen, an **A**utomated **T**ransformer **G**enerator for **C**ontrollable **Gen**eration. We instantiate ATGC-Gen with both decoder-only (e.g., GPT) and encoder-only (e.g., BERT) transformer architectures to examine their respective capabilities for controllable generation. It is designed to encode and integrate heterogeneous biological information, such as cell types, protein sequences, and transcription activation signals, into the sequence generation process. This unified framework enables ATGC-Gen to capture complex relationships between DNA sequences and their biological contexts, facilitating the generation of sequences with specific desirable properties.

To evaluate the effectiveness of our proposed approach, we apply ATGC-Gen, a transformer-based language model that incorporates cross-modal biological information for controllable DNA sequence generation. By conditioning on diverse properties such as cell types, transcription factors, and regulatory signals, ATGC-Gen enables flexible and biologically grounded sequence design. We first assess its performance on established promoter (Avdeyev et al., 2023; Stark et al., 2024; Sarkar et al., 2024) and enhancer (Taskiran et al., 2024) generation tasks. To further explore its capability in handling complex biological contexts, we introduce a new dataset based on ChIP-Seq experiments, which involves generating sequences that bind to specific proteins in specific cell types. Experimental results across different tasks, evaluated in terms of functionality, fluency, and diversity, demonstrate that ATGC-Gen achieves strong and consistent performance, highlighting the promise of language models for advancing controllable and property-aware DNA sequence design.

In summary, our main contributions are as follows:

- We propose **ATGC-Gen**, a language model framework for controllable DNA sequence generation. ATGC-Gen supports flexible conditioning on diverse biological modalities, enabling biologically meaningful sequence design under complex control constraints.
- We introduce a new dataset for controllable DNA generation based on ChIP-Seq experiments, which captures protein-DNA binding patterns. The benchmark includes well-structured evaluation metrics to assess functionality, fluency, and diversity.
- We conduct extensive experiments on promoter, enhancer, and ChIP-Seq-based tasks. Results demonstrate that ATGC-Gen outperforms strong baselines in generating accurate, coherent, and diverse sequences under various biological conditions.

2 Background and Related Work

DNA Generation. We study the problem of DNA sequence generation. In the field of synthetic biology and genetic engineering, unconditional or random DNA generation has limited applications, since DNA sequences are designed or edited to achieve specific outcomes, *e.g.,* producing a protein or altering a metabolic pathway. An unconditionally generated DNA sequence lacks the necessary context to be meaningful or useful in biological systems. Hence, in this work, we focus on studying the task of DNA generation given specific properties, such as a particular organism, cell type, or functional requirement. Formally, we define a DNA dataset as $\mathcal{U} = \{(U_1, C_1), (U_2, C_2), \cdots, (U_N, C_N)\}$, where U_i is a DNA sequence and C_i is its corresponding biological property. We aim to learn a conditional generative model $p_\theta(\cdot|C_i)$ on $\mathcal{U}$ so that the model G, parameterized with θ, can generate DNA sequences fulfilling C_i. Previous studies have made several initial attempts to apply deep learning methods for controllable DNA generation. The Dirichlet Diffusion Score Model (DDSM) (Avdeyev et al., 2023) uses Dirichlet distribution to discretize the diffusion process, effectively generating DNA sequences. In Stark et al. (2024) the Dirichlet distribution is used in the flow matching process, enhancing the quality of generated DNA sequences. DiscDiff (Li et al., 2024) employs Latent Discrete Diffusion model on the DNA generation task and proposes a post-training refinement algorithm to improve the generation quality. D3 (Sarkar et al., 2024) uses score entropy for discrete diffusion on the conditional DNA sequence generation.

DNA Language Model. Recent research has demonstrated the power of language models for learning from discrete sequences, which is ideally suited for modeling DNA sequences. A few recent studies have employed language models for encoding DNA sequences in prediction tasks. Early works adopt Transformers to encode DNA sequence data, and notable examples include DNABERT, DNABERT-2 and Nucleotide Transformer (Ji et al., 2021; Zhou et al., 2023b; Dalla-Torre et al., 2023). Recent works focus more on using state-space models, producing models like HyenaDNA (Nguyen et al., 2024b) and Mamba (Gu & Dao, 2023). HyenaDNA employs the Hyena operator (Poli et al., 2023) and uses implicit convolution to handle long DNA sequences up to one million bases effectively. Mamba captures long-range interactions in a simplified manner, demonstrating significant effectiveness in long sequence modeling (Gu & Dao, 2023; Schiff et al., 2024). Caduceus (Schiff et al., 2024) improves Mamba based on bi-directional encoding and reverse complement nature of DNA sequence. Both HyenaDNA and Mamba have been pretrained using the next

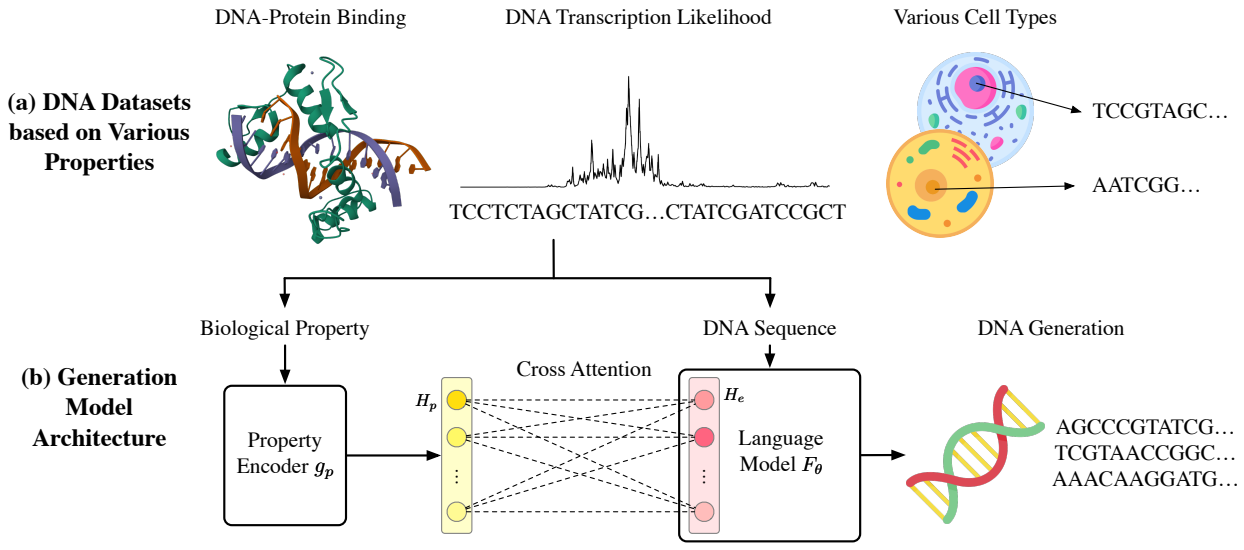

Figure 1: Overview of the proposed ATGC-Gen framework. (a) Different biological properties used for generating DNA sequences. (b) The architecture of the generation model.

token prediction paradigm, making them particularly well-suited for DNA generation tasks. Based on the state space model, Evo (Nguyen et al., 2024a) publishes a pretrained model encoding DNA, RNA, protein modalities, demonstrating impressive performance across a variety of tasks in these modalities. However, current studies invariably focus on DNA prediction tasks and do not involve generation. On the other hand, the true power of language models lies in their generation capabilities. Hence, our primary aim in this work is to apply language models for controllable DNA generation, generating sequences to satisfy predefined biological properties across various modalities.

3 The Proposed ATGC-Gen

In this section, we introduce the overall framework of **ATGC-Gen** for controllable DNA sequence generation, as illustrated in Figure 1. The framework encodes various biological property modalities and employs a language model to generate DNA sequences that are aligned with specified properties. We begin by describing how the property encoder is used to transform biological property inputs into dense representations. Next, we explain how these property representations are integrated with the DNA sequence embeddings to condition the language model. Finally, we present the training objective used to optimize the model.

3.1 Representation Encoding

In this section, we illustrate encoding method for target properties from different modalities.

Sequence-level Integration. Sequence-level integration provides a general mechanism for incorporating biological properties in a global form. This approach encodes the property as a set of global representations that summarize the desired biological context—such as cell types or pooled statistics.

Given a property matrix $\mathbf{P} \in \mathbb{R}^{l_p \times d_p}$, where l_p is the number of global property tokens and d_p is the input feature dimension, we transform it into the model's hidden space using a learnable linear projection $\mathbf{H}_p = \mathbf{P}W_p^\top + \mathbf{b}_p$, where W_p and b_p are learnable parameters.

Let the DNA embedding sequence be $\mathbf{H}_e = \{\mathbf{h}_{e,1}, \cdots, \mathbf{h}_{e,n}\} \in \mathbb{R}^{n \times d_h}$. We prepend the property embedding to the DNA sequence to form the final transformer input:

$$\mathbf{H}_{\text{input}} = \{\mathbf{H}_p; \mathbf{H}_e\} \in \mathbb{R}^{(l_p+n) \times d_h}. \tag{1}$$

This design allows the transformer to treat the property tokens as global control inputs, enabling them to influence generation via self-attention across the full input sequence.

Feature-level Integration. Feature-level integration can be applied when the property sequence is aligned position-wise with the DNA sequence, i.e., both have the same length. This setting is applicable to base pair resolution properties, such as transcriptional signals defined at each nucleotide position.

Let the DNA sequence be represented as a token sequence $U = \{u_1, \cdots, u_n\}$, where $u_i \in \{A, C, G, T\}$ representing the specific nucleotide. We convert it to one-hot embeddings $\mathbf{X}_e = \{\mathbf{x}_{e,1}, \cdots, \mathbf{x}_{e,n}\}$, where $\mathbf{x}_{e,i} \in \mathbb{R}^4$ is the one-hot encoding of u_i. Let the aligned property sequence be $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$ with each $\mathbf{s}_i \in \mathbb{R}^{d_s}$.

For each position i, we concatenate the DNA one-hot and property vectors:

$$\tilde{\mathbf{x}}_i = \text{concat}(\mathbf{x}_{e,i}, \mathbf{s}_i) \in \mathbb{R}^{4+d_s}.$$

We then apply a shared linear transformation to map the concatenated vector to the model's hidden dimension:

$$\tilde{\mathbf{h}}_{e,i} = W_f \tilde{\mathbf{x}}_i + \mathbf{b}_f,$$

where $W_f \in \mathbb{R}^{d_h \times (4+d_s)}$ and $\mathbf{b}_f \in \mathbb{R}^{d_h}$. The resulting transformer input is

$$\mathbf{H}_{\text{input}} = \{\tilde{\mathbf{h}}_{e,1}, \cdots, \tilde{\mathbf{h}}_{e,n}\} \in \mathbb{R}^{n \times d_h}. \tag{2}$$

Feature-level integration enables the model to condition on fine-grained biological signals at each position in the DNA sequence.

3.2 Training Objectives

We consider two alternative training paradigms for controllable DNA sequence generation: an *autoregressive* training objective based on decoder-only transformers (e.g., GPT), and a *masked language model* objective based on encoder-only transformers (e.g., BERT). Both approaches condition on external biological properties as described in the previous section.

Autoregressive Training. In this setting, we train a decoder-only language model F_θ to generate DNA sequences in an autoregressive way, conditioned on the property representations $\mathbf{H}_p$. The input sequence consists of the property tokens followed by the DNA tokens, and the model is trained to predict the next nucleotide at each position. This training paradigm naturally complements sequence-level integration, where the prepended property tokens act as a global prefix that controls all subsequent predictions.

Following standard practice (Zhou et al., 2023b; Schiff et al., 2024; Radford et al., 2018), we use a single-nucleotide tokenizer, where each base (A, C, G, T) is treated as a discrete token. The model is trained using the next-token prediction objective with cross-entropy loss:

$$\min_\theta \ \mathbb{E}_{U \sim \mathcal{U}} \left[\sum_{i=1}^{|U|-1} \ell \left(F_\theta \left(\mathbf{H}_p, u_1, \cdots, u_i \right), u_{i+1} \right) \right], \tag{3}$$

where $U = \{u_1, \cdots, u_{|U|}\}$ is a DNA sequence from the dataset $\mathcal{U}$, and ℓ is the cross-entropy loss over the predicted nucleotide distribution.

This autoregressive formulation naturally supports variable-length sequence generation, making it suitable for DNA design tasks with flexible output lengths.

Masked Language Modeling. In parallel, we consider an encoder-based training approach using masked language modeling (MLM), similar to BERT (Devlin et al., 2019). This formulation requires sequences of fixed length and is trained to recover randomly masked tokens in the input. Specifically, given an input

sequence of DNA tokens $U = \{u_1, \cdots, u_n\}$ and property representation $\mathbf{H}_p$, we randomly select a subset of positions $\mathcal{M} \subset \{1, \cdots, n\}$ to mask, and train the model F_θ to reconstruct the original nucleotides at those positions:

$$\min_{\theta} \ \mathbb{E}_{U \sim \mathcal{U}} \left[\sum_{i \in \mathcal{M}} \ell \left(F_\theta \left(\mathbf{H}_{\text{input}} \right)_i, u_i \right) \right], \tag{4}$$

where $\mathbf{H}_{\text{input}}$ is the full sequence embedding (property and DNA sequence with some tokens masked), and $F_\theta(\cdot)_i$ denotes the model prediction at position i.

Although the masked language model objective requires fixed-length sequences and is not naturally suited for generation, it enables efficient parallel training and leverages bidirectional context, making it expressive and effective in some certain cases.

3.3 Controllable Generation

Given a trained model and specified biological properties, we perform controllable DNA sequence generation by conditioning on the encoded property representations described earlier. We support two generation strategies, corresponding to the two training objectives: autoregressive decoding with a decoder-only (GPT-style) model, and masked language modeling with an encoder-only (BERT-style) model.

Autoregressive Generation. In the GPT-style setup, generation proceeds from left to right. At each time step i, the model predicts the next nucleotide u_i conditioned on the property representation $\mathbf{H}_p$ and previously generated tokens $u_1, \cdots, u_{i-1}$:

$$p_\theta(u_i \mid \mathbf{H}_p, u_1, \cdots, u_{i-1}).$$

We initialize generation with a special start token and stop when a designated end-of-sequence token is generated or a maximum sequence length is reached.

To promote diversity and controllability, we sample tokens using temperature sampling (Ackley et al., 1985; Ficler & Goldberg, 2017).

Masked Recovery Generation. For the BERT-style model, generation is performed by iterative or parallel masked token prediction. We initialize the entire sequence with masked tokens: $\{\texttt{[MASK]}, \cdots, \texttt{[MASK]}\}$, and jointly condition on the property representation $\mathbf{H}_p$. The model then predicts the nucleotide for each masked position randomly.

There are two generation modes in this setting:

- **One-shot unmasking:** all positions are predicted in parallel from a fully masked input in a single forward pass.
- **Iterative unmasking:** a subset of masked positions is predicted at each step, and the predictions are fed back in; the remaining positions stay masked and are updated progressively.

This formulation enables parallel generation and full bidirectional conditioning, but requires fixing the sequence length in advance.

Remark. This formulation shares structural similarities with recent masked denoising approaches, such as masked discrete diffusion (Sahoo et al., 2024; Nie et al., 2025), where a corrupted input is progressively reconstructed through iterative masked token prediction.

4 Controllable DNA Generation with ChIP-Seq

We develop a dataset derived from ChIP-Seq experiments for evaluating controllable DNA generation problems. ChIP-Seq experiments aim to identify the binding sites of DNA-associated proteins on the genome,

Table 1: Examples of the ChIP-Seq generation dataset.

Chrom	ChromStart	ChromEnd	Transcription Factor (TF)	Cell Type	Score
chr1	26677454	26677790	TCF7	K562	405
chr4	64158376	64159457	CTCF	medulloblastoma	1000
chr2	190070290	190070667	RAD21	GM12878	1000
chr11	86800296	86800780	MXI1	SK-N-SH	277

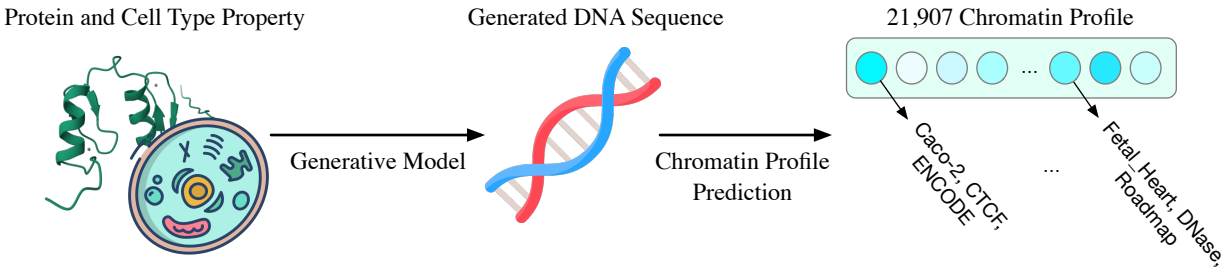

Figure 2: Illustration of the ChIP-Seq generation task. The objective is to generate DNA sequences based on specific proteins and cell types. The functionality of the generated DNA sequences is evaluated using chromatin profile prediction.

specifically the DNA sequences where proteins physically attach or bind. In our setting, we focus on predicting potential binding DNA sequences of variable lengths based on given protein sequences across different cell types. This prediction can advocate our understanding of the complex networks of gene regulation and aid in designing DNA sequences with specific functionalities for genetic engineering.

Our proposed dataset is obtained from a comprehensive collection of ChIP-Seq experiments generated by the ENCODE project (Consortium et al., 2012), which identifies specific DNA sequences to which proteins bind. We filter the raw data by considering the highest binding scores, length properties and other criteria, and detailed steps are provided in the Appendix A.

Dataset Description. Table 1 shows several example rows of the ChIP-Seq generation dataset. The raw dataset contains approximately 10 million rows, with each row describing a binding relationship between a DNA sequence and a protein for specific cell type(s). This dataset contains 340 transcription factors (proteins) and 129 cell types. In addition to the (DNA, protein, cell type) triplets, each row presents the binding scores to indicate the binding strength.

The DNA sequences are from GRCh38 (HG38) reference human genome assembly, used for mapping and aligning genetic data. Each base pair in the DNA sequence is one of the four nucleotides: adenine (A), cytosine (C), guanine (G), thymine (T). Given the protein name, we can query protein databases (Berman et al., 2000) to find the corresponding protein sequences and structures.

Task and Evaluation Metrics. The controlled generation task involves generating DNA sequences based on a given cell type and binding protein. Figure 2 illustrates the generation task. It is important to note that for the DNA-protein binding, a single protein may bind with different DNA sequences, and the same DNA sequence can bind to different proteins.

We propose three quantitative metrics to evaluate the generated sequences.

- **Functionality**: This metric measures how well the generated DNA sequence binds with the given protein under a specific cell type. We use the SEI framework (Chen et al., 2022), which employs a deep learning model to predict 21,907 chromatin profiles based on DNA sequences. As shown in Figure 2, each chromatin profile is defined by cell type, protein name and experiment. The prediction value indicates the probability

of binding between the DNA and the protein. We focus exclusively on chromatin profiles derived from ENCODE experiments, as our ChIP-Seq dataset is sourced from the ENCODE project.

- **Fluency**: Similar to the fluency used in the text generation task (Li et al., 2022; Zhou et al., 2023a), we adopt it to assesses how smooth and natural the generated DNA sequence is. Fluency is defined as the perplexity of the generated DNA in a pretrained language model, with lower value indicating better fluency. We use the HyenaDNA pretrained model as the base model F_b (Nguyen et al., 2024b). Given the cross-entropy loss ℓ, the fluency of a single DNA sequence $U = \{u_1, \cdots, u_n\}$ is the exponential of the loss

$$\text{Flu} = \exp(\sum_i \ell\left(F_b(u_1, \cdots, u_i), u_{i+1}\right)). \tag{5}$$

  The total fluency is then calculated as the mean fluency across all DNA sequences.

- **Diversity**: In a specific cell type, a given protein can bind to many different DNA sequences. This metric measures the diversity of the generated sequences. Following Li et al. (2024), the diversity score is defined as

$$\text{Diversity} = \sum_c (W_c \times \prod_{n=10}^{12} \frac{|\text{count}(n, U)|}{\text{count}(n, U)}), \tag{6}$$

  where $\text{count}(n, U)$ is the total number of n-grams in the generated DNA sequence U, and $|\text{count}(n, U)|$ is the number of unique n-grams. Given the variability in the number of samples in each category c, the weight of diversity in each category is denoted by $W_c = N_c/N$, where N_c is the number of samples in category c and N is the total number of samples.

5 Experiments

5.1 Promoter Generation

Dataset and task descriptions. This dataset contains 100,000 promoter sequences from GRCh38 (HG38) human reference genome. Each sequence is annotated with the transcription initiation profile, indicating the likelihood of transcription initiation activity at each base pair (fan, 2014). We use the same DNA sequences as in previous works (Avdeyev et al., 2023; Stark et al., 2024), specifically splitting 1,024 base pair long DNA sequences around each annotated transcription start site position (Hon et al., 2017). The task is to interpret the transcription initiation profile into the corresponding DNA sequences of the same length.

Metrics. Following prior works (Avdeyev et al., 2023; Stark et al., 2024), we evaluate the generation performance using the mean squared error (MSE) between the predicted regulatory activity of the generated and original DNA sequences. The regulatory activity is computed using the SEI framework (Chen et al., 2022), a published deep learning model that predicts 21,907 regulatory features for a given DNA sequence. For our primary evaluation, we extract promoter-related activity by focusing on the H3K4me3 chromatin mark. In addition, we report the Kolmogorov–Smirnov (KS) statistic between the distributions of regulatory activity values from the generated and original sequences, as proposed in (Sarkar et al., 2024), to assess distributional alignment. We further evaluate the fluency of the generated sequences by HyenaDNA (Nguyen et al., 2023), as described in Section 4, to ensure that generated DNA is biologically plausible and structurally coherent.

Results and analysis. Table 2 shows the generation performances for the promoter sequences. The results show that **ATGC-Gen-BERT** achieves the best overall performance compared to previous baselines, demonstrating the strong modeling capacity of the masked recovery generation paradigm. In contrast, **ATGC-Gen-GPT** aggregates base-resolution activity features at the beginning of the sequence, which removes per-token alignment between the activity signals and the DNA sequence. This loss of fine-grained information leads to inferior performance in functional metrics. However, due to its autoregressive nature, ATGC-Gen-GPT better preserves the fluency and coherence of the generated DNA sequences.

Table 2: Performance on promoter design. The best performance is indicated in bold, while the second-best performance is underlined. This convention is followed in all subsequent tables.

Method	MSE ↓	KS Test Statistic ↓	Fluency ↓
BIT Diffusion	0.0395	-	-
D3PM-UNIFORM	0.0375	-	-
DDSM	0.0334	-	-
Dirichlet FM	0.0269	0.399	3.6538
D3	<u>0.0219</u>	0.052	<u>3.5255</u>
ATGC-Gen-GPT	0.0289	<u>0.048</u>	**3.5231**
ATGC-Gen-Bert	**0.0192**	**0.043**	3.5904

Table 3: Performance on enhancer generation.

	Fly Brain			Melanoma		
Method	FBD ↓	Diversity ↑	Fluency ↓	FBD ↓	Diversity ↑	Fluency ↓
Dirichlet FM	1.0404	**0.8314**	4.0512	1.9051	**0.8395**	3.7126
ATGC-Gen-GPT	**0.5080**	0.8309	**4.0326**	**0.9228**	0.8131	**3.6852**

5.2 Enhancer Generation

Dataset and task descriptions. This task involves two distinct datasets: one from fly brain (Janssens et al., 2022) and the other from human melanoma cells (Atak et al., 2021). We follow the same data split as in the work by Stark et al. (2024), using 104k fly brain DNA sequences and 89k human melanoma DNA sequences, each consisting of 500 base pairs. The cell class labels are derived from the ATAC-seq experiments (Buenrostro et al., 2013), with 47 classes for the human melanoma dataset and 81 classes for the fly brain dataset. Our objective is to generate DNA sequences based on the given cell class labels.

Metrics. Following Stark et al. (2024), we evaluate the performance using the Fréchet Biological Distance (FBD) between the generated and original DNA sequences. To calculate the FBD score, we use a pretrained classifier model (Stark et al., 2024) to obtain the hidden representations and then compute the Wasserstein distance between Gaussians. Additionally, we assess performance using the diversity score and fluency score, similar to the metrics proposed in Section 4. These metrics provide a comprehensive evaluation of the generated DNA sequences.

Results and analysis. Table 3 presents the results for enhancer DNA sequence generation. Our proposed **ATGC-Gen-GPT** achieves a significantly lower Fréchet Biological Distance (FBD) than baseline models, indicating its effectiveness in incorporating global property information and modeling cross-modality signals to generate biologically realistic sequences. In terms of generation quality, ATGC-Gen-GPT produces more fluent and coherent sequences than the flow matching-based model, although with reduced diversity. Additional results for **ATGC-Gen-BERT** are provided in Appendix C.2. Compared to the GPT variant, ATGC-Gen-BERT demonstrates weaker performance in capturing global property information, likely due to the limitations of the masked modeling objective in autoregressive-style generation.

5.3 ChIP-Seq Generation

Results and Analysis. Table 4 reports the results on the ChIP-Seq generation task, described in Section 4. Since the task involves variable-length generation, we only evaluate **ATGC-Gen-GPT**. The model achieves a strong Binding Score when conditioned on biological properties. It also achieves higher diversity compared to other ablations. The performance drop without property inputs highlights the importance of conditioning on biological context. Nonetheless, the property-free model still performs better than random sequences, suggesting that ATGC-Gen captures meaningful sequence patterns even without explicit control signals.

Table 4: Performance on ChIP-Seq DNA generation. "w/" indicates that DNA sequences are generated based on the biological properties, while "w/o" indicates that the properties are not provided during generation.

Method	Binding Score ↑	Diversity ↑
Real Sequence	0.2319	0.0513
Random Sequence	0.0036	-
ATGC-Gen w/o Property	0.0747	0.1213
ATGC-Gen w/ Property	**0.1176**	**0.1228**

6 Conclusion and Discussion

In this paper, we present **ATGC-Gen**, a controllable DNA sequence generator that integrates diverse biological properties via cross-modal encoding and leverages language models for conditional generation. Our method enables accurate, fluent, and biologically relevant sequence design across multiple tasks. To support this direction, we also construct a new dataset from ChIP-Seq experiments, providing a realistic benchmark for DNA-protein binding generation.

Limitations. Despite promising results, our work is constrained by limited computational resources, preventing the training of larger models. Moreover, frequent evaluations during training, especially under autoregressive settings, introduce notable computational overhead.

Broader Impacts. Controllable DNA generation offers exciting opportunities for synthetic biology and genetic design. While it may accelerate the development of beneficial traits, it also raises concerns about misuse or the unintended synthesis of harmful sequences. Ensuring biological safety and ethical safeguards is critical.

Future Directions. Future work may explore more advanced generation techniques (e.g., retrieval-augmented models), as well as the development of richer, biologically grounded benchmarks to drive progress in programmable genomics.

References

A promoter-level mammalian expression atlas. *Nature*, 507(7493):462–470, 2014.

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Zeynep Kalender Atak, Ibrahim Ihsan Taskiran, Jonas Demeulemeester, Christopher Flerin, David Mauduit, Liesbeth Minnoye, Gert Hulselmans, Valerie Christiaens, Ghanem-Elias Ghanem, Jasper Wouters, et al. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. *Genome research*, 31(6):1082–1096, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL `https://doi.org/10.1093/nar/28.1.235`.

Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, 10(12):1213–1218, 2013.

Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022.

ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pp. 2023–01, 2023.

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

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*, 2017.

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

Chung-Chau Hon, Jordan A Ramilowski, Jayson Harshbarger, Nicolas Bertin, Owen JL Rackham, Julian Gough, Elena Denisenko, Sebastian Schmeier, Thomas M Poulsen, Jessica Severin, et al. An atlas of human long non-coding rnas with accurate 5
textquotesingle ends. *Nature*, 543(7644):199–204, 2017.

Jasper Janssens, Sara Aibar, Ibrahim Ihsan Taskiran, Joy N Ismail, Alicia Estacio Gomez, Gabriel Aughey, Katina I Spanier, Florian V De Rop, Carmen Bravo Gonzalez-Blas, Marc Dionne, et al. Decoding gene regulation in the fly brain. *Nature*, 601(7894):630–636, 2022.

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

Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing dna with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Shuiwang Ji. Dynamic search for inference-time alignment in diffusion models. *arXiv preprint arXiv:2503.02039*, 2025.

Zehui Li, Yuhao Ni, William AV Beardall, Guoxuan Xia, Akashaditya Das, Guy-Bart Stan, and Yiren Zhao. Discdiff: Latent diffusion model for dna sequence generation. *arXiv preprint arXiv:2402.06079*, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

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

Eric Nguyen, Michael Poli, Matthew G Durrant, Armin W Thomas, Brian Kang, Jeremy Sullivan, Madelena Y Ng, Ashley Lewis, Aman Patel, Aaron Lou, et al. Sequence modeling and design from molecular to genome scale with evo. *bioRxiv*, pp. 2024–02, 2024a.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024b.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Anirban Sarkar, Ziqi Tang, Chris Zhao, and Peter Koo. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv*, 2024.

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.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Xingyu Su, Haiyang Yu, Degui Zhi, and Shuiwang Ji. Learning to discover regulatory elements for gene expression prediction. *arXiv preprint arXiv:2502.13991*, 2025.

Ibrahim I Taskiran, Katina I Spanier, Hannah Dickmänken, Niklas Kempynck, Alexandra Pančíková, Eren Can Ekşi, Gert Hulselmans, Joy N Ismail, Koen Theunis, Roel Vandepoel, et al. Cell-type-directed design of synthetic enhancers. *Nature*, 626(7997):212–220, 2024.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and dna design. *arXiv preprint arXiv:2502.14944*, 2025.

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.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pp. 42602–42613. PMLR, 2023a.

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

Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K Speicher, Christoph S Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, et al. Controlling gene expression with deep generative design of regulatory dna. *Nature communications*, 13(1):5099, 2022.

A Data Processing for ChIP-Seq Experiment

Based on the data from the ChIP-Seq experiment, we perform data preprocessing. No license is needed for the data files and database tables[1]. The raw data comes from the ENCODE project[2]. The preprocessing steps are as follows:

Filtering by Binding Scores. Binding score measures how effectively DNA sequences bind to proteins. To focus on the most relevant interactions, we retain only the highest binding scores for each category. This step ensures that our analysis concentrates on the strongest and most meaningful DNA-protein interactions.

Alignment with SEI Framework. To ensure compatibility with the SEI framework used for downstream evaluation of DNA sequence functionality, we align the cell types and protein names in our dataset with those recognized by SEI. Specifically, we retain the GM12878 cell type, which has the largest number of entries. Additionally, we filter the dataset to include only proteins whose names are supported by the SEI framework, enabling accurate and consistent evaluation.

Standardizing Input Sequence Lengths. The raw dataset contains pairs of DNA sequences and their associated binding transcription factor (TF) protein sequences, with lengths varying significantly, ranging from 50 to over 20,000 base pairs for DNA and up to over one thousand amino acids for proteins. Such variation poses challenges for modeling and computation. To ensure consistency and tractability, we filter the dataset to retain only DNA sequences shorter than 500 base pairs and protein sequences shorter than 1,000 amino acids. This standardization facilitates efficient training and stable convergence of the language models.

Embedding Protein Sequences. To represent the transcription factor (TF) proteins, we use the ESM-2-3B protein language model (Lin et al., 2023) to extract contextual embeddings from the amino acid sequences. We retain the full-length output embeddings without applying any sequence-level pooling or aggregation, thereby preserving residue-level resolution for downstream tasks.

After completing the preprocessing steps, we partition the dataset by chromosome: chromosomes 20 and 21 are used for validation, while chromosomes 22 and X are held out for testing. The remaining chromosomes are used for training. The detailed statistics of the cleaned and processed dataset are provided in Table 5.

Table 5: ChIP-Seq dataset information after preprocessing.

# rows	# proteins	# split samples
55830	62	51800/2181/1849

B Implementation Details

We report the hyperparameters and training configurations used in our experiments. For **ATGC-Gen-Bert**, we adopt a BERT-style encoder with the `bert-base` configuration: 12 layers, 12 attention heads, and a hidden size of 768. For **ATGC-Gen-GPT**, we use a GPT-style decoder with 16 layers, 16 attention heads, and the same hidden size of 768. We use the AdamW optimizer with a learning rate of 1×10^{-4} and a linear warmup over the first 10% of training epochs. Model selection is based on performance on the validation set. The batch size is adjusted to fit within GPU memory constraints. All training and inference are conducted on NVIDIA A100-SXM-80GB GPUs.

Table 6: Performance on enhancer generation without property information.

Method	Fly Brain			Melanoma		
	FBD ↓	Diversity ↑	Fluency ↓	FBD ↓	Diversity ↑	Fluency ↓
Dirichlet FM w/o Property	15.2107	**0.0697**	4.0847	**5.3874**	**0.0696**	3.8251
ATGC-Gen-GPT w/o Property	**14.6412**	0.0572	**4.0229**	8.3796	0.0579	**3.6329**

Table 7: Performance between ATGC-Gen-Bert and ATGC-Gen-GPT on enhancer generation.

Method	Fly Brain			Melanoma		
	FBD ↓	Diversity ↑	Fluency ↓	FBD ↓	Diversity ↑	Fluency ↓
ATGC-Gen-GPT	0.5080	0.8309	4.0326	0.9228	0.8131	3.6852
ATGC-Gen-Bert	27.63	0.8667	3.9609	45.27	0.8230	3.7560

C Additional Results

C.1 Enhancer Design without Property Information

To better understand the role of conditioning signals, we follow prior work and evaluate ATGC-Gen-GPT in an uncontrolled setting, where no property information is provided during generation (Stark et al., 2024). As shown in Table 6, ATGC-Gen-GPT results in a mixed performances compared with the Dirichlet FM baseline in this scenario. This result contrasts with the controllable generation setting, where ATGC-Gen-GPT consistently outperforms Dirichlet FM by a large margin (Table 3). The discrepancy highlights that ATGC-Gen is particularly effective when it can leverage external biological properties to guide sequence generation. In the absence of such conditioning, its advantage diminishes. Overall, these findings suggest that while Dirichlet FM remains competitive in unconditional generation, ATGC-Gen is better suited for property-conditioned tasks—precisely the setting that is most relevant in practical genomic design applications.

C.2 Enhancer Design with ATGC-Gen-Bert

Table 7 presents a comparison between ATGC-Gen-BERT and ATGC-Gen-GPT on the enhancer generation tasks. Despite both variants using the same sequence-level integration to encode global property information, ATGC-Gen-BERT performs substantially worse than ATGC-Gen-GPT in terms of Fréchet Biological Distance (FBD). This performance gap suggests that masked recovery generation, as used in ATGC-Gen-BERT, is less effective at capturing and utilizing global properties compared to the autoregressive decoding in ATGC-Gen-GPT. In contrast, the autoregressive nature of GPT allows for sequential incorporation of such information, leading to better alignment with the desired biological conditions.

C.3 Effect of Unmasking Rate on ATGC-Gen-Bert

During the generation process of ATGC-Gen-Bert, we can control the number of tokens unmasked at each step. For instance, we may choose to unmask all masked tokens simultaneously or unmask only one randomly selected token per step. Figure 3 presents the results under varying unmasking rates, where the x-axis denotes the number of unmasking steps normalized by sequence length (e.g., 1.0 corresponds to unmasking one token per step for a 1024-length sequence). The figure shows that overall performance remains relatively stable across different unmasking schedules. Notably, using fewer steps (e.g., unmasking 10% of tokens per step, or 103 steps in total) yields comparable performance to full-length generation (1024 steps), suggesting the potential for significant speedup. However, the extreme case of unmasking all tokens in a single step results in a notable performance drop (MSE = 0.0334), indicating the importance of stepwise refinement. In Table 2, we report results using the most conservative setting, unmasking one token per step. While not the best-

[1] https://genome.ucsc.edu/license/

[2] https://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg38&g=encRegTfbsClustered

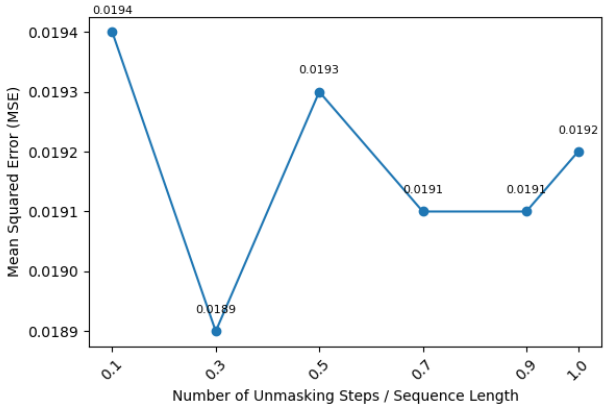

Figure 3: Effect of unmask ratio on ATGC-Gen-Bert in promoter generation.

performing configuration in Figure 3, we adopt this setting to avoid selecting hyperparameters based on testing set performance, which could introduce bias.