Unlocking Efficiency: Adaptive Masking for Gene Transformer Models

Soumyadeep Roy[a,*]**, Shamik Sural**[a] **and Niloy Ganguly**[a]

[a]Indian Institute of Technology Kharagpur
ORCID (Soumyadeep Roy): https://orcid.org/0000-0001-7269-2163, ORCID (Shamik Sural):
https://orcid.org/0000-0002-4315-7329, ORCID (Niloy Ganguly): https://orcid.org/0000-0002-3967-186X

Abstract. Gene transformer models such as Nucleotide Transformer, DNABert, and LOGO are trained to learn optimal gene sequence representations by using the *Masked Language Modeling* (MLM) training objective over the complete Human Reference Genome. However, the typical tokenization methods employ a basic sliding window of tokens, such as k-mers, that fail to utilize gene-centric semantics. This could result in the (trivial) masking of easily predictable sequences, leading to inefficient MLM training. Time-variant training strategies are known to improve pretraining efficiency in both language and vision tasks. In this work, we focus on using curriculum masking where we systematically increase the difficulty of *masked token prediction* task by using a *Pointwise Mutual Information*-based difficulty criterion, as gene sequences lack well-defined semantic units similar to words or sentences of NLP domain. Our proposed *Curriculum Masking-based Gene Masking Strategy* (CM-GEMS) demonstrates superior representation learning capabilities compared to baseline masking approaches when evaluated on downstream gene sequence classification tasks. We perform extensive evaluation in both few-shot (five datasets) and full dataset settings (*Genomic Understanding Evaluation* benchmark consisting of 27 tasks). Our findings reveal that CM-GEMS outperforms state-of-the-art models (*DNABert-2, Nucleotide transformer, DNABert*) trained at 120K steps, achieving similar results in just 10K and 1K steps. We also demonstrate that Curriculum-Learned LOGO (a 2-layer DNABert-like model) can achieve nearly 90% of the state-of-the-art model performance of 120K steps. We will make the models and codes publicly available at https://github.com/roysoumya/curriculum-GeneMask.

1 Introduction

The field of developing large foundational models in genomics is gaining popularity [7]; these models are primarily trained on the Human Reference Genome, such as DNABert, LOGO, and Nucleotide Transformer (Human Reference Genome model variant). These models use Masked Language Modeling (MLM), pioneered by BERT [10], where the model predicts original tokens of a masked subset as clues. The state-of-the-art pretrained models (DNABert [11] and LOGO [30]) in gene sequence classification tasks such as *promoter region prediction, core promoters, enhancers, splice sites prediction, and functional genetic variants*, are widely

* Corresponding Author. Email: soumyadeep.roy9@iitkgp.ac.in
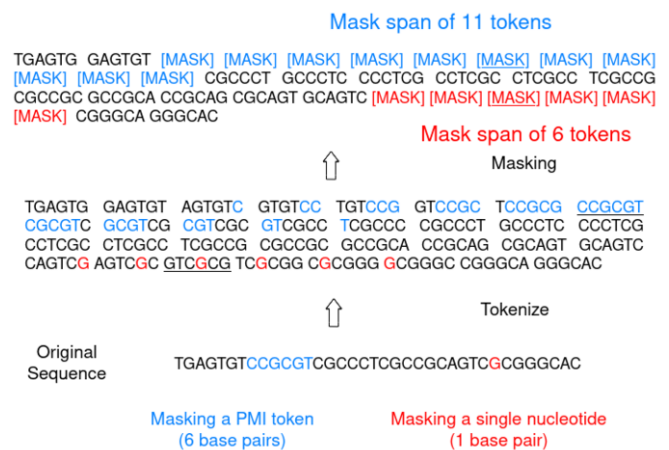
Figure 1. Masked language modeling pipeline of gene transformer models. The original sequence is broken into k-mers (here, k = 6). The difference between masking a single nucleotide versus a PMI token is highlighted by using red and blue font, respectively

used in literature [6, 16]. The recently introduced *Genome Understanding Evaluation* [35] (GUE) benchmark covers these gene sequence modeling tasks for multiple species such as *human, mouse, yeast,* and *virus*; this serves as a standard resource for evaluating the gene foundational models. Here, each task takes as input a sequence of nucleotides that consists of bases such as adenine (A), cytosine (C), guanine (G), and thymine (T), and an associated class label. The gene transformer models are first pretrained using the masked language modeling (MLM) objective to learn optimal gene sequence representations. However, unlike the NLP domain, where well-defined semantic units called words or sentences exist, no well-defined semantically demarcated tokens exist for a given sequence.

Therefore, to come to a workable solution, researchers randomly select a sequence of k nucleotides [11, 16]; we use $k = 6$ throughout this study, following prior works [21, 11, 30]. Figure 1 clearly depicts converting a gene sequence into a sequence of k-mers. For example, if 'CCGCGT' is a frequently occurring sequence, then if we mask just one nucleotide within, it may be easily predicted based on its highly co-occurring nucleotides. Therefore, the model can easily predict the masked token by learning trivial patterns, leading to increased pretraining time. DNABert requires a total time of 25 days to complete pretraining for 120K steps on 8 NVIDIA

2080Ti GPUs. Consens et al. [7] mentioned this requirement of huge compute as a key limitation of current gene foundational models, and approximates the cost of running the popular models of DNABERT [11], Enformer [4], Nucleotide Transformer [9] and HyenaDNA [18], at a cost of $5000 USD on 8 A100 GPUs. Therefore, we believe efficient and fast pretraining of gene foundational models is an important research area. This work focuses on fast pretraining of gene transformer models such as DNABert and LOGO.

Anagnostidis et al. [3] discusses the concept of 'compute-optimal' training and shows that adaptive training strategies significantly reduce the required compute to reach a given target performance compared to their static counterparts. We aim to achieve equivalent performance for a specified model (gene transformers in our case) with fewer computational resources than initially projected. We can develop an efficient pretraining strategy whereby we mitigate the instances of 'easy' learning by masking entire spans of highly correlated spans that occur frequently. This prevents the model from wasting its pretraining steps on predicting such trivial cases. Such intelligent masking schemes [14, 12] have been popular in the NLP domain. Roy et al. [21] developed the GENEMASK masking strategy and showed such a reduction in the number of pretraining steps, from 120K steps to 10K steps in the case of DNABert. This paper shows two reduction schemes: (i) further reduction in pretraining steps - 10K to 1K, and (ii) reduction in model size in terms of layers or transformer blocks. (12 to 2).

Roy et al. [21] utilize the Pointwise Mutual Information (PMI) score to prioritize highly correlated spans within a gene sequence instead of the random span masking strategy as used in the case of DNABert (or LOGO). PMI is computed as the probability of a set of tokens occurring together compared to appearing independently throughout the corpus. A high PMI score indicates a high correlation. Specifically, GENEMASK [21] first randomly selects certain positions as mask centers and then selects the highest-ranked PMI token present within a fixed neighborhood of that mask center. However, this strategy potentially misses the top-ranked PMI tokens in a gene sequence and may instead mask lower-ranked PMI tokens, thus not fully utilizing the PMI information. This may allow further wastage of pretraining steps and thus result in a delayed convergence. To mitigate this issue, we develop GLOBAL masking strategy that always captures the top-m highest ranked PMI tokens in a sequence, and thus ensures that the PMI information is strictly used, and may potentially lead to faster convergence and better few-shot performance. We observe that GLOBAL outperforms all the baseline models at 10000 steps when tested in the few-shot setting (combined 10 and 50-shot).

Few studies advocate using dynamic pretraining strategies, which is slowly garnering research interest. Yang et al. [29] is the first work to explore time-variant MLM settings into account; they develop two strategies called *Masking Ratio Decay* and *POS-Tagging Weighted Masking*, where they adaptively tune the masking ratio and masked content as the pretraining progresses. Combining the idea of curriculum masking, we use PMI to regulate/decide the difficulty level, whereby during the pretraining step, tokens for easier concepts are masked first. In contrast, in the later part, the harder concepts are used. Therefore, contrary to the time-invariant masking strategy of GENEMASK (i.e., stays fixed during the entire pretraining procedure), we develop a dynamic (time-variant) masking strategy that updates the masking algorithm (and consequently the masking ratio) during pretraining. Our proposed method CM-GEMS achieves better few-shot performance than competing baseline models over the five standard gene sequence classification datasets (gene regulatory elements such as promoters, enhancers, silencers, and

splice sites). We also perform extensive ablation analysis and experiments to better understand the efficacy of CM-GEMS. We make the codes and data publicly available at https://github.com/roysoumya/curriculum-GeneMask.

In this work, we make the following contributions:

1. To the best of our knowledge, CM-GEMS is the first work to apply curriculum masking and time-variant (dynamic) pretraining strategy for training gene transformer models. CM-GEMS trained for 10K steps outperforms the state-of-the-art models trained for 120K as well as the state-of-the-art gene masking algorithm, GENEMASK, by 2.18% and 1.24% respectively on average across both 'Human' and 'non-Human' species of the GUE benchmark [35].
2. We develop the GLOBAL masking strategy that directly (globally) selects the top-ranked PMI tokens throughout the entire gene sequence (of 512 tokens) and outperforms GENEMASK by 2.66% on average across five datasets and their corresponding 10 and 50-shot settings.
3. We perform extensive evaluation over 32 datasets (5 few-shot and 27 full training data) across two gene transformer model architectures of DNABert and LOGO. Although Gene Mask showed improvement only in terms of few-shot performance, we also show results on the full dataset (GUE benchmark) and observe that CM-GEMS at 1K steps and GLOBAL at 10K can achieve performance within a 90% margin than state-of-the-art models such as DNABert-2 and Nucleotide transformers trained for 120K steps.

2   Related Works

Here, we summarize the recent works on gene sequence modeling, followed by various masking strategies used for masked language modeling (MLM) based pretraining strategies, including time-variant pretraining and curriculum masking.

Representation learning for genomics. Badirli et al. [6] represent the DNA barcodes as a vector representation for fine-grained species classification. Gene sequences are typically represented as k-mers by recent works [17, 11, 30], which is then used to learn a dense representation from an adapted BERT model. However, other works such as BigBird [32] and DNABert-2 [35] use a *byte-pair encoding* scheme to build the model vocabulary. Mo et al. [16] develop a multimodal pretrained model using a combination of gene sequences and transcription factors. There is a recent line of work to learn long-range interactions present in the human genome: BigBird [32], Enformer [4] and HyenaDNA [18]. Given the context length limitation of transformer architectures, Avsec et al. [4] utilizes a combination of convolutional and transformer layers to encode the long sequences of one-hot encoded base pairs into a more condensed vector representation. However, the *Enformer* model focuses on gene expression track prediction, which is not the focus of this work. In this work, we focus on gene sequence classification tasks.

MLM masking strategies in NLP. BERT [10] apply *random token masking* where 15% of the input tokens are chosen to be masked uniformly. Whole word masking [25] and Entity masking [26] show performance improvement over the standard 'random token masking' scheme. Here, they masked a contiguous sequence of tokens that either formed a whole word (*whole word masking*) or an entity (*entity masking*). However, a more straightforward approach called 'random span masking' outperformed entity masking, where the mask centers are randomly selected, and the span length w.r.t this mask center is drawn from a geometric distribution. However, there are no

pre-defined semantics similar to entities and words of NLP, which can be applied to gene sequences; therefore, it is impossible to apply entity and whole-word masking in our case. Levine et al. [14] use the Pointwise Mutual Information (PMI) score to identify the sequence of contiguous tokens to be masked together. Such intelligent masking shows accelerated performance whereby they achieve end-of-pretraining performance at almost half the training steps. Sadeq et al. [22] proposes the 'Informative Relevance' metric, which, for a given word, is computed as the sum of PMI values between a masked word and all unmasked words in the given sentence. However, the same issue is the lack of gene semantics, where the equivalent of a natural language sentence is not known for the gene sequence modeling domain.

Time-variant pretraining and curriculum masking. Curriculum learning is also used in reinforcement learning framework [31]. Curriculum masking easy-to-hard schedule for masked image modeling using vision transformer blocks [15]. Pavlova and Makhlouf [20] used easy-to-hard curriculum masking for biomedical named entity recognition tasks. The notion of task difficulty involves (i) masking words instead of tokens and (ii) increasing the masking rate. Yang et al. [29] demonstrate that using time-invariant pretraining, particularly in terms of masking ratio and masked content, does not produce good results. They propose a novel masked content selection strategy based on the parts of speech of a given word; masking non-function words proved to be more useful than masking function words. Lee et al. [13] introduced the idea of curriculum masking (easy-to-hard) where they first map words or phrases to a concept in an existing knowledge graph called Concept-Net. They define words and phrases related to many other concepts as 'easy concepts' and use them to initialize the curriculum masking schedule. As the pretraining progresses, it gradually masks concepts related to the previously masked concepts during the consecutive stages.

However, in our case, we neither have any word or phrase-level gene semantics nor access to such a large, well-annotated knowledge graph. Lee et al. [13] applied it in the NLP domain where they evaluated the General Language Understanding Evaluation (GLUE) benchmark that comprises single sentence or sentence pair classification tasks. Lee et al. [13] found that the hard-to-easy curriculum performs poorly and leads to a drop in performance, which they explain may be because the hard concepts at the start proved to be too difficult without prior knowledge or learning from relevant, easier concepts. Therefore, in our proposed time-variant masking strategy, CM-GEMS, we initially adopt an easier masking strategy of GENEMASK, followed by a hard masking strategy, GLOBAL that only masks the top-ranked PMI tokens.

3 Research Background

Here, we introduce the two popular gene transformer models of DNABert [11] and LOGO [30] that are based on the BERT architecture. These models learn a robust contextual representation of DNA sequence fragments by leveraging unsupervised training based on Masked Language Modeling over the Human Reference Genome; it contains 3.2 billion nucleotides over 24 chromosomes. We will explain the complete pipeline by mapping to the stages similar to pre-trained language models (PLMs) developed for natural language processing tasks. We divide it into two phases: (i) preprocessing and (ii) pretraining. We use the same experimental design and training strategy as used by Roy et al. [21] for evaluating the different masking strategies for fast pretraining of gene transformer models.

3.1 Preprocessing

Tokenization of gene sequences. Here, we represent a gene sequence as a sequence of k-mers, as used by prior studies [11, 17] while developing gene transformer models. k-mer represents a sliding window of length k, as depicted in Figure 2. We adopt the value of k as 6, following the setup used by Roy et al. [21]. 6-mers are more appropriate because they incorporate richer contextual information and have a manageable memory and compute requirement [30].

Pretraining data preparation. We use the pretraining data constructed and made available by Roy et al. [21] to maintain fair comparison. Next, we summarize their pretraining data construction steps. We obtain the *Human Reference Genome* data in the *FASTA* format from the *Genome Reference Consortium Human Build 38 patch release 13* (GRCh38.p13) [8]. This dataset comprises long, unlabeled gene sequences divided across multiple chromosomes and is used by gene transformer models to perform unsupervised training using masked language modeling loss. We do not allow sequences that contain bases apart from that of A, T, C, or G. The sequence length is determined as 510 token long for half the cases and a random length between 5 and 510 for the remaining half.

3.2 Pretraining

The standard masked language modeling (MLM) loss, same as that used for BERT [10], is used as a pretraining objective for training DNABert and LOGO. A key difference between the NLP domain with that of gene transformers is that to mask a single nucleotide, one has to mask a continuous sequence of tokens to avoid trivial masked token prediction. This is because a single nucleotide is present as part of k contiguous k-mers. In a more formal context, let us denote a nucleotide as DNA[i], while representing a 6-mer token as T[i], which is equivalent to DNA[i-2], DNA[i - 1] $\cdots$ DNA[$i + 3$]. Consequently, the tokens T[j], $\forall (j)_{j=i-2}^{i+3}$ are subject to masking. Given a token length of $k = 6$ and the requirement to mask 15% of tokens [10], the probability for masked language model (MLM) is established at $15\%/6$, resulting in 2.5% of nucleotides chosen for masking (given our primary focus on 6-mers unless specifically indicated as T[i] = $T_6[i]$).

3.3 GENEMASK: *State-of-the-art Masking Strategy*

GENEMASK [21] is the state-of-the-art time-invariant masking algorithm for Masked Language Model (MLM) training of gene transformer models. It showed improvement over the random span masking strategy used by DNABert and LOGO by masking all the nucleotides simultaneously in the most correlated spans (determined using *Pointwise Mutual Information value, PMI* [14]). Masking correlated spans helps to reduce information leakage and enhances the model's ability to predict intricate patterns. However, retaining the advantages of the conventional random masking approach is also necessary. Therefore, GENEMASK first randomly selects positions within a gene sequence as mask centers, which is then followed by locally selecting top-ranked *normalized PMI* (NPMI$_k$, see Equation 1) within a fixed neighborhood of the mask center, for masking purposes. The formal definition of Normalized Point-wise Mutual Information (NPMI$_k$), where k > 2, as proposed by Roy et al. [21].

$$\text{NPMI}_k(w_1 \ldots w_k) = PMI_k * \frac{\log f(w_1 \ldots w_k)}{\log(c) + \log f(w_1 \ldots w_k)} \quad (1)$$

$f(w_1 \ldots w_k)$ represents the number of times the k-mer sequence of $w_1 \ldots w_k$ occurs. The minimum frequency of occurrence is referred to as c, which is used to filter rare tokens.

4 Proposed Methodology

Our proposed masking strategy for fast pretraining is called *Curriculum Masking-based Gene Masking Strategy* (CM-GEMS). Optimizing the Pointwise Mutual Information (PMI) score prioritizes the masking of k (=6)-mers that co-occur more frequently than their components. It takes full advantage of the existing GENEMASK strategy (Section 3.3). It later shifts to our proposed GLOBAL masking strategy (Section 4.1) towards the end of the pretraining stage. Our key innovation lies in adopting a time-variant masking strategy that automatically transitions to a new masking strategy during the pretraining process when certain conditions (in this case, perplexity score) are satisfied. At the start of the pretraining, all tokens are unfamiliar to the models. However, as the pretraining progresses, the relative predictive difficulty of each token will vary. We also adopt an easy-to-hard curriculum masking schedule based on consistent observations of prior works [13, 29].

4.1 Global PMI Masking Strategy

We always mask the highest PMI tokens instead of locally optimal PMI tokens as done for GENEMASK. The GENEMASK masking strategy of random masking for 50% of cases and locally optimal PMI masking for the remaining 50% of cases is replaced by masking only the highest-ranked PMI tokens. Specifically, we select the top m nucleotides as mask centers (MC) with the highest $NPMI_k$ score. We term this masking strategy as 'Global Masking Strategy' (GLOBAL), and it is further explained in detail in Algorithm 1.

Determining the value of m: A PMI token corresponds to one 6-mers, which contains six nucleotides. Therefore, we need to mask a span of 11 adjacent tokens to mask one PMI token because of six mask centers, two left-side tokens, and three tokens towards the right. The original MLM probability for masking one nucleotide was $15\%/6 = 2.5\%$. In our case, it is gets reduced to $15\%/11 = 1.36\%$. Given the input sequence length is 512 tokens, $m \approx 7$.

Algorithm 1: GLOBAL Algorithm

Input: Input sequence of 6-mer tokens having a maximum of 510 tokens, Pre-computed Normalized PMI$_k$ (NPMI$_k$) values for all 6-mers stored as a dictionary

Output: $MaskTokenSet$: Token indices within input sequence to be masked

Initialization: // A 6-mer token present at i-th position is represented as T[i], the i-th nucleotide is represented as DNA[i]

$MaskTokenSet \leftarrow \emptyset$

T[i] $\leftarrow$ {DNA[i-2] $\cdots$ DNA[i+3]}

Function MapNucleotideToKmerTokens (*nucleotide position id i*):

   | $MappedTokens \leftarrow$ T[j], $\forall(j)_{j=i-2}^{i+3}$
 | **return** $MappedTokens$

Step 1: Sort the DNA string with 6-mer tokens in a non-increasing order of $NPMI_k$ score.

Step 2: Create a priority set with the top-ranked m nucleotides with the highest $NPMI_k$ score.

Step 3: for *each nucleotide in priority set* **do**

   | // Masking a PMI token involves masking 11 adjacent tokens
   | $MaskTokenSet \leftarrow MaskTokenSet \cup \forall(j)_{j=\tau-2}^{\tau+3}$
   | $MapNucleotideToKmerTokens(\texttt{j})$

return $MaskTokenSet$
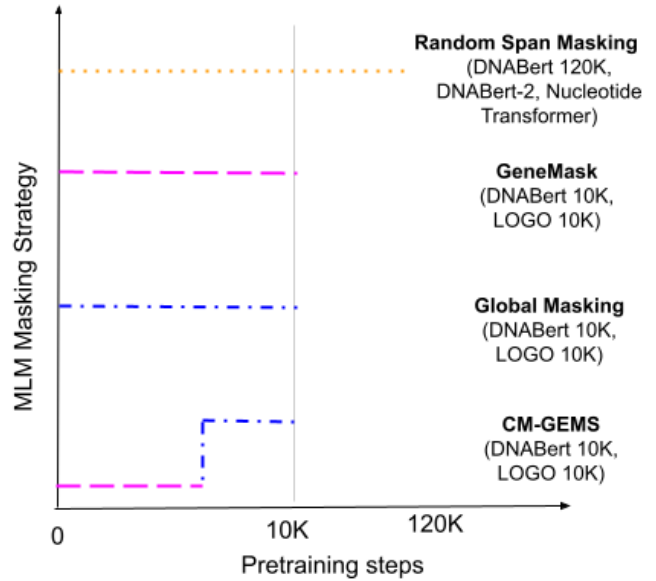
Figure 2. Comparison of CM-GEMS and time-invariant strategy of GLOBAL with existing masking strategies

4.2 CM-GEMS: *Curriculum Masking-based Gene Masking Strategy*

Motivation. The GENEMASK masking strategy limits its search for the highest PMI tokens within the local context of the randomly selected mask centers. Although this enforces the PMI tokens being uniformly selected, GENEMASK may miss the top-ranked PMI tokens globally, i.e., within the entire sequence of 512 tokens. The top-ranked PMI tokens are the most difficult to predict and force the model to learn deeper patterns that eventually lead to faster convergence or pretraining. Since GENEMASK may miss masking the top-ranked PMI tokens, it may lead to trivial masking and delayed convergence. We develop CM-GEMS to address this limitation in this work.

Proposed Masking Algorithm of CM-GEMS. CM-GEMS takes advantage of both GENEMASK algorithm proposed by Roy et al. [21] as well as the GLOBAL algorithm proposed in this paper. We formulate it as a two-stage easy-to-hard curriculum masking strategy, where GENEMASK acts as the 'easy' stage of the curriculum. We allow the GENEMASK masking strategy till the model continues to learn, which we measure by the drop in perplexity score. Once the drop in perplexity score falls below a threshold value (in our case, the value is one), we consider that the model improves marginally from this stage onwards. Therefore, we then shift to the GLOBAL masking strategy, which is the 'hard' stage of the curriculum, as it only considers the top-ranked PMI tokens. The only associated change to the pretraining hyper-parameters, apart from the masking algorithm, is the change in masking rate, which is controlled by the 'mlm_probability' field; it reduces from 1.765% of GENEMASK to 1.36% of GLOBAL. Figure 2 depicts the two-stage curriculum masking strategy of CM-GEMS. To the best of our knowledge, this is the first work to explore the time-variant pretraining strategy for fast pretraining of genome foundational models.

Dataset	Task	Num. of classes	Sequence length	Train / Test
Prom-core	Promoter	2	70	53276 / 5920
Prom-300	Promoter	2	300	53276 / 5920
Cohn-enh	Enhancer	2	500	20843 / 6948
Splice-40	Splice Sites	3	40	24300 / 3000
Sil-300	Silencer	2	300	14909 / 1657

Table 1. Descriptive statistics of few-shot evaluation datasets

5 Experimental Setup

We describe the datasets and the evaluation setup, followed by the training details for CM-GEMS and baseline models.

5.1 Datasets

Full dataset evaluation. We use the Genome Understanding Evaluation (GUE) benchmark recently introduced by DNABERT-2 [35] for evaluating in the full dataset setting. It consists of 7 gene sequence classification tasks with 28 datasets with input sequence lengths ranging between 70 to 1000; we could not evaluate on *Covid Variant Classification* task (one dataset) as it was not included in their evaluation scripts. Therefore, we finally tested on 27 datasets, where the number of target classes is two (binary classification) except for *Splice Site Detection*, which has three target classes. Unlike the few-shot evaluation setup described above that only focuses on human species, 15 out of the 27 datasets of GUE belong to Non-human species (5 from Mouse for the *Transcription Factor Prediction* task and 10 from Yeast for the *Epigenetic Marks Prediction* task). This is an important additional feature that further helps us evaluate the multi-species transferability in the 'genome understanding' capability of each model.

Few-shot evaluation. We evaluate across five gene sequence classification tasks, where the input is a gene sequence, i.e., a sequence of nucleotides, and the task is to predict whether it is a gene regulatory element or not. The promoter region prediction tasks (Prom-core, Prom-300) and Enhancer-Cohn prediction task (Cohn-enh) are obtained from Roy et al. [21]. We constructed the 'silencer' dataset: the positive data points are obtained from the 'SilencerDB' database [33], and used the 'high-throughput Homo-sapiens' version. We followed the same negative set construction technique as done for Prom-core and Prom-300, as well as for DeePromoter [19].

5.2 Evaluation Setup

We use the same metric for performance comparison, as mentioned in the respective prior works, to maintain a fair comparison. We report the statistical significance results based on paired t-test for the performance improvement by CM-GEMS over baseline models.

Full dataset evaluation on GUE. We use the *Matthews Correlation Coefficient* (MCC) evaluation metric as used by Zhou et al. [35]. *MCC* is used to measure the binary classification quality where it takes into account true and false positives and negatives. 'MCC' can take values between -1 and 1, where $+1$ indicates an exact match, and -1 represents a complete disagreement between the ground truth and prediction.

Few-shot evaluation. We report accuracy as used by Roy et al. [21] for few-shot performance comparison of gene sequence classification tasks. The experiments are repeated for ten times by selecting a random seed and a set of n data points for each class, corresponding to each run; the mean and standard deviation values are reported.

We only show results for values of $n = 10, 50$, i.e., the 10-shot and 50-shot settings respectively. We do not perform any hyperparameter tuning as access to a validation dataset is not assumed, following the prior few-shot classification works [24, 23, 21].

5.3 Training Details

Full dataset evaluation on GUE. We directly use the evaluation script provided by Zhou et al. [35] in their GitHub codebase for evaluating our proposed models. We use the performance scores reported by the authors of DNABert-2 and the Nucleotide transformer.

Few-shot evaluation. We follow the same training details as the work of Roy et al. [21] for setting the GeneMask, ORI10K results, i.e., we use the same hyperparameter values as the original state-of-the-art (SoTA) models, except for 'warmup steps' and 'maximum pretraining steps' (see Section 5 of Roy et al. [21] to know more about training and fine-tuning) details, as well as the hyperparameters used. We follow the same setup for the remaining model variants for the different pretraining step variants at 1000 and 10000.

5.4 Baseline Models

We evaluate the impact of the different masking strategies on two gene transformer architectures of DNABert [11] and LOGO [30].

State-of-the-art (SoTA) Models. The recent gene transformer models such as DNABERT-2 [35] and Nucleotide transformers [9] do not make their pretraining codes publicly available, making it not possible to test the impact of our proposed masking strategy on such models. For a fair comparison, we select variants of SoTA models that are trained only on the Human Reference Genome and use the k-mer tokenization scheme (k=6). 'SoTA-Best' represents the best result among the three k-mer-based SoTA models, such as DNABert 120K, NT-500M-Human, and DNABert-2 120K trained on 120K steps.

Time-invariant Masking Strategies. The models are pretrained for 10K steps using the same hyperparameter settings as Roy et al. [21] to obtain a fair comparison. The original SoTA models, such as DNABert and LOGO, use random span masking. We use the pretrained weights of the DNABert model, which was trained for 120K steps, and use it as a baseline model referred to as *ORI 120K* model.

Time-variant Masking Strategies. We introduce one time-variant, curriculum masking-based baseline model called **CM-Step** where we start with the 100% of random masking strategy (p_{random}), with a uniform increase in the percentage of global PMI strategy (p_{GLOBAL} starting at 0%); $p_{random} + p_{GLOBAL} = 100\%$. We first divide the maximum pretraining steps into blocks of 100 (as p_{GLOBAL} needs to increase from 0% to 100%. We draw from a binomial distribution where the number of trials (n) is one and the probability of success (p) is p_{GLOBAL} for selecting the masking strategy.

6 Experimental Results

We show the performance comparison of our proposed masking algorithm, CM-GEMS in Table 2 with the SoTA gene foundational models (DNABert-2, Nucleotide Transformer), the SoTA masking strategy for gene transformers (GENEMASK), and few baseline masking approaches. CM-GEMS outperforms GENEMASK in both the full dataset setting on the GUE benchmark (see RQ1) as well as the few-shot setup (see RQ3). In this paper, we explore the following research questions:

RQ1: CM-GEMS evaluation on the full dataset with GUE benchmark. We observe that our proposed curriculum-trained

| Task | Dataset | Models | | | | | | | | | | | |
| | | State-of-the-art models | | | | DNABert | | | | LOGO | | | |
		DNABert-2 120K (BPE)★	DNABert 120K◇	NT-500M-Human◇	DNABert-2 120K (k-mer)★	Global 10K	GeneMask 10K	CM-Step 10K	CM-GEMS 10K	Global 10K	GeneMask 10K	CM-Step 10K	CM-GEMS 10K
		Species: Human											
PD	all	85.57	90.48	87.71	83.78	89.93	89.50	90.48	89.29	85.82	82.93	85.88	84.76
(Human)	no tata	92.55	93.05	90.75	92.65	91.09	91.73	91.83	92.41	89.52	88.54	89.79	88.62
	tata	60.85	61.56	78.07	57.75	76.25	79.76	81.12	77.47	68.24	69.79	69.31	72.21
CPD	all	66.28	68.90	63.45	74.91	70.56	68.54	71.65	72.09	69.16	64.13	67.34	63.66
(Human)	notata	67.99	70.47	64.82	69.23	70.87	70.01	72.13	70.18	68.91	67.29	66.84	66.23
	tata	72.73	76.06	71.34	74.91	76.96	74.95	75.51	83.50	70.85	53.65	55.55	61.71
TFP	0	66.99	66.84	61.59	67.99	65.89	67.40	65.44	66.07	64.21	67.09	63.94	65.34
(Human)	1	70.98	70.14	66.75	67.06	71.14	69.81	68.35	68.95	69.47	67.85	67.49	69.75
	2	61.40	61.03	53.58	59.45	57.66	59.80	58.61	57.66	53.37	55.31	53.83	55.31
	3	55.10	51.89	42.95	50.24	46.80	51.26	47.65	51.41	40.20	42.48	44.70	40.49
	4	71.31	70.97	60.81	72.80	74.10	76.15	73.22	72.60	70.32	70.54	68.65	69.98
Splice	Reconstruct	79.62	84.07	79.71	77.90	83.02	84.84	84.74	84.12	77.92	74.01	80.25	75.20
Mean (Human)		70.95	**72.12**	68.46	70.72	72.86	73.65	73.39	**73.81**	**69.00**	66.97	67.80	67.77
		Species: Non-human											
EMP	H3	77.08	73.10	69.67	74.62	71.45	73.28	74.07	74.35	64.72	60.90	61.91	61.49
(Yeast)	H3K14ac	55.60	40.06	33.55	42.71	38.75	40.73	40.27	41.28	30.38	32.55	33.34	29.49
	H3K36me3	57.25	47.25	44.14	47.26	44.11	45.42	46.06	46.83	38.94	39.26	38.52	35.92
	H3K4me1	45.51	41.44	37.15	39.66	41.63	42.36	40.91	44.61	31.19	28.66	31.04	25.38
	H3K4me2	40.83	32.27	30.87	25.33	30.60	33.14	31.40	33.43	30.99	29.32	30.11	27.11
	H3K4me3	42.57	27.81	24.00	27.43	25.91	25.92	24.93	30.24	18.34	15.35	22.65	12.22
	H3K79me3	66.01	61.17	58.35	61.03	59.18	60.45	59.20	59.83	54.36	52.19	55.70	53.38
	H3K9ac	56.79	51.22	45.81	49.35	49.24	52.22	51.77	52.77	43.54	42.16	45.62	40.04
	H4	80.07	79.26	76.17	78.61	76.38	76.04	75.83	76.66	72.81	66.51	71.51	68.85
	H4ac	54.19	37.43	33.74	37.14	33.89	37.43	35.69	36.21	27.76	27.84	31.84	27.5
TFP	0	48.01	44.42	31.04	48.96	49.48	52.57	50.48	54.32	12.02	27.93	27.16	42.23
(Mouse)	1	81.86	78.94	75.04	81.69	79.70	79.05	79.90	80.51	71.30	69.73	70.93	69.50
	2	82.98	71.44	61.67	81.71	75.50	78.08	74.40	80.70	52.50	59.80	55.57	78.19
	3	73.22	44.89	29.17	63.17	51.00	60.27	52.51	61.01	34.87	48.96	34.38	61.53
	4	46.15	42.48	29.27	42.83	41.04	42.60	42.68	44.94	21.40	28.78	25.89	26.60
Mean (Non-human)		60.54	51.55	45.31	**53.43**	51.19	53.30	52.01	**54.51**	40.34	42.00	42.41	**43.96**

Table 2. Performance comparison in terms of Matthews Correlation Coefficient (MCC) metric on the Genome Understanding Evaluation (GUE) benchmark. However, for the masking baselines, we pretrain for 10000 steps, following the evaluation setup prescribed by Roy et al. [21]. The task names are shortened due to space constraints - Promoter Detection (PD), Core Promoter Detection (CPD), Transcription Factor Prediction (TFP), Splice Site Prediction (Splice) and Epigenetic Marks Prediction (EMP). ◇ and ★ indicate that the values are directly taken from Tables 6 and 10 of Zhou et al. [35]. The statistical significance test results using paired t-test are provided in Table 3

Models	GUE Dataset	p-value
DNABert 120K	60.69 ± 17.72	0.021
DNABert-2 120K (k-mer)	61.12 ± 17.77	0.043
GENEMASK 10K	62.34 ± 17.69	0.112
CM-GEMS 10K	63.09 ± 17.49	

Table 3. Performance comparison of CM-GEMS with state-of-the-art models on complete GUE benchmark. Statistical significance results using 'paired t-test' are also provided. p-value ≤ 0.05 indicates that the performance improvement of CM-GEMS is statistically significant

model, *DNABert* CM-GEMS *10K*, achieves the best performance in both Human and Non-human tasks, outperforming both the k-mer-based SoTA models as well as the best masking strategy-based baseline model (GM 10K). *LOGO* CM-GEMS *10K*, a 15 million parameter model (compared to 117 million parameters of DNABert-2, achieves 93.97% of the best SoTA model's performance (based on *Matthews Correlation Coefficient* score), using just 10K steps instead of 120K. Table 3 shows the statistical significance test results where we observe that the performance improvement of CM-GEMS over DNABert 120K and DNABert-2 120K is statistically significant, unlike GENEMASK that fails to outperform DNABert 120K by a significant margin. Although CM-GEMS improves over GENEMASK by 1.20%, the improvement is not statistically significant.

RQ2: CM-GEMS evaluation at reduced compute and model size with GUE benchmark data. We evaluate the effectiveness of our

model trained with curriculum masking by only pretraining for 10% steps (10% of 10000 = 1000 steps), i.e., at just 1000 steps. Figure 3 shows the performance comparison of two such settings: (i) we keep the same architecture, which is DNABert, and (ii) we reduce the number of transformer blocks (12 to 2) by evaluating with the LOGO model. We observe that CM-GEMS 1K outperforms GENE-MASK 1K models in the case of DNABert and LOGO for the 'Human' species-related tasks of the GUE benchmark data. However, in the case of tasks related to the 'Non-human' species, we observe a reverse trend.

We also explore the perspective of 'intrinsic dimensionality' used in recent works [1, 34] to better understand fine-tuning phenomena in our case. *Intrinsic dimensionality* is defined as the minimum number of model parameters required to achieve 90% of the original model performance. Based on Figure 3, we observe that in the case of both Human and Non-human species, CM-GEMS achieves 90% of the SoTA-best 120K model, except for LOGO 1K for Non-human species, i.e., it satisfies the criterion for intrinsic dimensionality. This brings up an interesting research direction for building smaller, energy-efficient models that achieve similar performance.

RQ3: CM-GEMS evaluation in few-shot setting. We show the few-shot evaluation results in Table 4. Here, we observe that our proposed curriculum-trained model, CM-GEMS outperforms the current state-of-art masking strategy for gene transformers called GEN-EMASK (GM 10K) in terms of mean accuracy over five datasets by

Model Type	Prom-core		Prom-300		Cohn-enh		Splice-40		Sil-300		Mean
	10	50	10	50	10	50	10	50	10	50	
DNABert 120K	0.606 ± 0.045	0.687 ± 0.024	0.638 ± 0.070	0.808 ± 0.019	0.582 ± 0.030	0.638 ± 0.020	0.404 ± 0.019	0.472 ± 0.048	0.711 ± 0.062	0.849 ± 0.020	0.640
DNABert 10K	0.586 ± 0.051	0.653 ± 0.058	0.601 ± 0.065	0.789 ± 0.059	0.579 ± 0.047	0.634 ± 0.031	0.409 ± 0.017	0.512 ± 0.014	0.688 ± 0.068	0.823 ± 0.031	0.627
GM 10K	0.602 ± 0.058	0.678 ± 0.026	0.625 ± 0.09	0.781 ± 0.097	0.622 ± 0.050	0.648 ± 0.016	0.392 ± 0.02	0.505 ± 0.018	0.715 ± 0.061	0.818 ± 0.029	0.639
Global 10K	0.605 ± 0.05	0.691 ± 0.022	0.664 ± 0.076	0.834 ± 0.024	0.605 ± 0.047	0.627 ± 0.04	0.406 ± 0.021	0.514 ± 0.014	0.768 ± 0.062	0.849 ± 0.011	**0.656**
CM-Step 10K	0.579 ± 0.049	0.675 ± 0.022	0.649 ± 0.091	0.799 ± 0.034	0.608 ± 0.043	0.649 ± 0.017	0.402 ± 0.02	0.557 ± 0.022	0.706 ± 0.044	0.835 ± 0.023	0.646
CM-GEMS 10K	0.594 ± 0.052	0.684 ± 0.023	0.629 ± 0.075	0.822 ± 0.02	0.6 ± 0.059	0.645 ± 0.033	0.394 ± 0.018	0.516 ± 0.019	0.714 ± 0.057	0.844 ± 0.012	0.644

Table 4. Few-shot evaluation of CM-GEMS and baseline masking strategies in terms of accuracy across 10 and 50-shot settings with the base model being DNABert [11]; the evaluation setup is taken from Roy et al. [21], with the addition of the silencer task. We observe that CM-GEMS 10K outperforms GM 10K and the DNABert 120K model; Global 10K achieves the best performance.
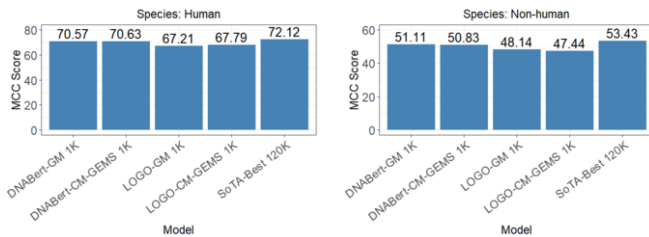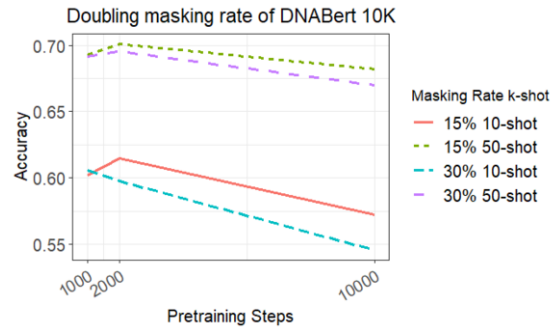

Figure 3. Performance comparison on GUE benchmark at reduced compute 1K pretraining steps and reduced model size (LOGO). (left) Human species, (right) Non-human species. The SoTA models are pretrained for 120K steps on the Human Reference Genome

Figure 4. Impact of doubling the masking rate of DNABert 10K model. The performance stays almost the same at 1K steps, but then it slowly decreases at 2K and 10K steps

a small margin of 0.78% (0.639 versus 0.644). We also observe that the Global 10K model achieved the best performance (mean accuracy of 0.656) among the baseline models, highlighting that such an only-hard-masking strategy proves useful in few-shot settings. However, based on our observation of Global 10K on the full dataset setting on the GUE benchmark as evident in Table 2, Global 10K performs poorly, we thus conclude that the efficacy of such a masking scheme is limited to few-shot settings.

RQ4: Impact of doubling the masking rate of DNABert model in a few-shot setting. Wettig et al. [28] showed in the NLP domain that both Random Span masking and PMI masking achieve the best downstream task performance (in both GLUE and SQuAD benchmark datasets) when the masking rate is kept at 30% (please see Figure 5 of Wettig et al. [28]). Therefore, we test the hypothesis of whether trivially doubling the masking rate leads to a performance improvement of the same order as that achieved by PMI-based masking strategies. We chose a few-shot setting (10 and 50-shot) for this purpose to notice the impact of this change in pretraining configuration by limiting the effect of huge amounts of fine-tuning data. Figure 4 shows that at the early stages of pretraining (i.e., at 1000 steps), there is a marginal difference, but as we progress further (at 2000 and 10000 steps), the performance (mean accuracy over five datasets) of DNABert with double the masking rate goes on decreasing. The performance drop at 10K steps over the standard masking rate of 15% is 4.71% and 1.79% respectively. The reason may be that with a higher masking rate, the model sees a much-reduced amount of unmasked or actual tokens that hinder the learning process.

7   Conclusion

We develop a novel (time-variant, easy-to-hard) curriculum masking algorithm, CM-GEMS for optimal masked language modeling training of gene transformer models. We also develop a time-invariant masking strategy called GLOBAL, where we specifically prioritize the top-ranked PMI tokens, which we refer to as the 'hard' strategy of our proposed curriculum masking strategy. Our study highlights the limitations of conventional tokenization methods in gene trans-

former models and proposes a novel curriculum masking approach to address these shortcomings. By systematically increasing the difficulty of the masked token prediction task based on Pointwise Mutual Information, we enhance the representation learning capabilities of Gene Transformer models such as DNABert and LOGO. CM-GEMS outperforms state-of-the-art models on average, in both few-shot gene sequence classification tasks (5 datasets) as well as on the Genomic Understanding Evaluation benchmark (27 datasets) trained on the full dataset. We also show results on the full dataset (GUE benchmark) and observe that CM-GEMS at 1K steps and GLOBAL at 10K can achieve performance within a 90% margin than state-of-the-art models such as DNABert-2 and Nucleotide transformers trained for 120K steps. This shows the potential of adaptive masking strategies for fast pretraining and effectively reducing the compute requirement for training gene foundational models, a major limitation of current models as pointed out by Consens et al. [7].

Future Work. Given the significant performance gain observed for DNA foundational models due to intelligent (PMI-based) masking strategies, we will extend our work to other foundational models based on genomics data, specifically RNA-based models such as CodonBERT [5], RNABERT [2] and single-cell RNA-based models such as GeneFormer [27]. Although PMI indirectly captures DNA sequence motifs [21], a much-needed inter-disciplinary research direction is to involve more biologically grounded pretraining or fine-tuning objectives instead of MLM [7]. We will perform a more systematic exploration of the 'intrinsic dimensionality' to better understand the fine-tuning dynamics of gene transformer models; to the best of our knowledge, this is a vastly unexplored domain with very limited research. We make the codebase available at https://github.com/roysoumya/curriculum-GeneMask.

Acknowledgements

References

[1] A. Aghajanyan, S. Gupta, and L. Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online, Aug. 2021. doi: 10.18653/v1/2021.acl-long.568.

[2] M. Akiyama and Y. Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics*, 4(1):lqac012, 02 2022. ISSN 2631-9268. doi: 10.1093/nargab/lqac012.

[3] S. Anagnostidis, G. Bachmann, and T. Hofmann. Navigating scaling laws: Accelerating vision transformer's training via adaptive strategies. *arXiv preprint arXiv:2311.03233*, 2023.

[4] Ž. Avsec, V. Agarwal, et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, Oct 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01252-x.

[5] A. N. Babjac, Z. Lu, and S. J. Emrich. Codonbert: Using bert for sentiment analysis to better predict genes with low expression. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '23, 2023. ISBN 9798400701269. doi: 10.1145/3584371.3613013.

[6] S. Badirli, Z. Akata, et al. Fine-grained zero-shot learning with dna as side information. In *Advances in Neural Information Processing Systems*, volume 34, pages 19352–19362, 2021.

[7] M. E. Consens, C. Dufault, M. Wainberg, et al. To transformers and beyond: Large language models for the genome. *arXiv preprint arXiv:2311.07621*, 2023.

[8] G. R. Consortium. Genome reference consortium human build 38 patch release 13 (grch38.p13), 2019. URL https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39/.

[9] H. Dalla-Torre, L. Gonzalez, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01, 2023.

[10] J. Devlin, M.-W. Chang, et al. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1423.

[11] Y. Ji, Z. Zhou, et al. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083.

[12] M. Joshi, D. Chen, et al. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 01 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00300.

[13] M. Lee, J.-H. Park, et al. Efficient pre-training of masked language model via concept-based curriculum masking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7417–7427, Abu Dhabi, United Arab Emirates, Dec. 2022. doi: 10.18653/v1/2022.emnlp-main.502.

[14] Y. Levine, B. Lenz, et al. {PMI}-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3Aoft6NWFej.

[15] N. Madan, N.-C. Ristea, K. Nasrollahi, et al. Cl-mae: Curriculum-learned masked autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2492–2502, January 2024.

[16] S. Mo, X. Fu, et al. Multi-modal self-supervised pre-training for regulatory genome across cell types. *CoRR*, abs/2110.05231, 2021. URL https://arxiv.org/abs/2110.05231.

[17] P. Ng. dna2vec: Consistent vector representations of variable-length k-mers. *CoRR*, abs/1701.06279, 2017. URL http://arxiv.org/abs/1701.06279.

[18] E. Nguyen, M. Poli, M. Faizi, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems*, volume 36, pages 43177–43201, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/86ab6927ee4ae9bde4247793c46797c7-Paper-Conference.pdf.

[19] M. Oubounyt, Z. Louadi, H. Tayara, and K. T. Chong. Deepromoter: Robust promoter predictor using deep learning. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00286.

[20] V. Pavlova and M. Makhlouf. BIOptimus: Pre-training an optimal biomedical language model with curriculum learning for named entity recognition. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 337–349, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.bionlp-1.31.

[21] S. Roy, J. Wallat, S. S. Sundaram, et al. GENEMASK: fast pretraining of gene sequences to enable few-shot learning. In *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 2002–2009, 2023. doi: 10.3233/FAIA230492.

[22] N. Sadeq, C. Xu, and J. McAuley. InforMask: Unsupervised informative masking for language model pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5866–5878, Abu Dhabi, United Arab Emirates, Dec. 2022. URL https://aclanthology.org/2022.emnlp-main.395.

[23] T. Schick and H. Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online, Apr. 2021. doi: 10.18653/v1/2021.eacl-main.20.

[24] T. Schick and H. Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021. doi: 10.18653/v1/2021.naacl-main.185.

[25] R. Sennrich, B. Haddow, et al. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*, 2016. doi: 10.18653/v1/p16-1162.

[26] Y. Sun, S. Wang, et al. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[27] C. V. Theodoris, L. Xiao, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, Jun 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06139-9.

[28] A. Wettig, T. Gao, Z. Zhong, and D. Chen. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia, May 2023. doi: 10.18653/v1/2023.eacl-main.217.

[29] D. Yang, Z. Zhang, and H. Zhao. Learning better masking for better language model pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7255–7267, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.400.

[30] M. Yang, L. Huang, et al. Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *Nucleic Acids Research*, 50(14):e81–e81, 05 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac326.

[31] A. Y. Yasutomi and T. Ogata. Automatic action space curriculum learning with dynamic per-step masking. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–7, 2023. doi: 10.1109/CASE56687.2023.10260397.

[32] M. Zaheer, G. Guruganesh, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297, 2020.

[33] W. Zeng, S. Chen, X. Cui, X. Chen, Z. Gao, and R. Jiang. SilencerDB: a comprehensive database of silencers. *Nucleic Acids Research*, 49(D1):D221–D228, 10 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa839.

[34] Z. Zhang, B. Liu, and J. Shao. Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1701–1713, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.95.

[35] Z. Zhou, Y. Ji, W. Li, P. Dutta, R. V. Davuluri, and H. Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oMLQB4EZE1.