Parameter-Efficient Fine-Tune on Open Pre-trained Transformers for Genomic Sequence

Huixin Zhan
Division of Artificial Intelligence in Medicine
Cedars-Sinai Medical Center
Los Angeles, CA, USA, 90048
Huixin.Zhan@cshs.org

Zijun (Frank) Zhang*
Division of Artificial Intelligence in Medicine
Cedars-Sinai Medical Center
Los Angeles, CA, USA, 90048
zijun.zhang@cshs.org

Abstract

Lately, pre-trained foundation models (PFMs) in DNA have achieved notable advancements in unraveling the linguistic nuances of the genome. As these foundational models expand in parameters and the number of downstream genomic tasks increases, Parameter-Efficient Fine-Tuning (PEFT) has become the de facto approach to fine-tune PFMs while decreasing the computational costs. Low-rank adapters and adaptive low-rank adaptation (AdaLoRA) are popular PEFT methods that introduce some learnable truncated singular value decomposition modules for efficient fine-tuning. However, both methods are deterministic, i.e., once a singular value is pruned, it stays pruned throughout the fine-tuning process. Consequently, deterministic PEFTs can underperform if the initial states, before pruning, are suboptimal—a challenge frequently encountered in genomics due to data heterogeneity. To address this issue, we propose an AdaLoRA with random sampling (AdaLoRA+RS) to prune and stochastically reintroduce pruned singular vectors, adhering to a cubic budget schedule. We evaluate the AdaLoRA+RS on PFMs within genome domain, DNABERT 1/2 and Nucleotide Transformer; and language domain, open pre-trained transformers (OPT). Our AdaLoRA+RS approach demonstrates performance ranging from slightly above to on par with the Full-Model Fine-Tuning (FMFT) across 13 genomic sequence datasets on two genome understanding tasks, while using less than 2% of the trainable parameters. For instance, in the human promoter detection, OPT-350M with AdaLoRA+RS achieves a 4.4% AUC increase compared to its FMFT baseline, leveraging only 1.8% of the trainable parameters. Our proposed AdaLoRA+RS provides a powerful PEFT strategy for modeling genomic sequence.

1 Introduction

DNA-centric pre-trained foundation models (PFMs), such as DNABERT (Ji et al., 2021), DNABERT-2 (Zhou et al., 2023), and Nucleotide Transformer (NT) (Dalla-Torre et al., 2023), have made significant progress in decoding the linguistic intricacies of the genome. An important paradigm of utilizing such PFMs is "pretraining+finetuning", i.e., pre-training on large-scale unlabeled genomic sequences, and then adaption to a particular genome understanding task. As models grow in size, the practice of full-model fine-tuning (FMFT), which involves retraining every parameter, becomes less practical. There are two lines of solutions to address this: first, model compression; second, parameter-efficient fine-tuning (PEFT). While model compression approaches are well-established in recent years, implementing them on large language models can be very expensive, as these techniques

*Corresponding author.

NeurIPS 2023 Generative AI and Biology Workshop.

typically necessitate FMFT (Ma et al., 2023). As a countermeasure, PEFTs fine-tune the model on only a small number of additional parameters, significantly decreasing the computational costs.

Both low-rank adapters (LoRA) (Hu et al., 2021) and adaptive low-rank adaptation (AdaLoRA) (Zhang et al., 2023) are popular methods in PEFTs. LoRA keeps the main pretrained weights of the model frozen and performs fine-tune on some additional LoRA blocks. While LoRA blocks are parameter-efficient, the number of ranks in each block is fixed and cannot be modified after training. This means that any adjustment to the rank necessitates retraining the blocks from the beginning. To address this issue, AdaLoRA adaptively decreases the total rank of all LoRA blocks and keeps the important singular values in each block according to their importance score. However, both LoRA and AdaLoRA operate deterministically. The deterministic approaches in PEFTs can result in suboptimal outcomes, particularly when the pre-pruning states were not ideal. To address this, we introduce AdaLoRA with random sampling (AdaLoRA+RS) that not only prunes but also probabilistically restores pruned singular vectors following a cubic budget schedule. AdaLoRA+RS adeptly retains crucial singular values, taking into account both their importance and sensitivity during current batch training. We test the AdaLoRA+RS on PFMs in genome domain, DNABERT 1/2 and Nucleotide Transformer; and on PFMs in language domain, open pre-trained transformers (OPT), for two genome understanding tasks, i.e., epigenetic marks prediction (EMP) and promoter detection (PD). Our AdaLoRA+RS method achieves performance that varies between slightly superior and equivalent to FMFT on 13 genomic sequence datasets, utilizing under 2% of the trainable parameters. Moreover, when fine-tuning OPTs using their specialized byte-level BPE (BBPE) encoding, their performance is on par with that of fine-tuned DNA-centric PFMs. This sheds light on the potential for broader applications of language-based PLMs.

In summary, our findings are as follows: **(1)** We introduce AdaLoRA with random sampling (AdaLoRA+RS), an enhancement of the traditional PEFTs that dynamically prunes and probabilistically restores pruned singular vectors, ensuring optimized performance by balancing the importance and sensitivity of different singular values during batch training. **(2)** The proposed AdaLoRA+RS approach for OPT attains performance on the Pareto front compared to its FMFT baseline with only 0.94% of the trainable parameters. **(3)** We find that PLMs with its tailored BBPE encoding could have a broader use cases. Notably, this encoding is well-suited for genomics data. We evaluate the AdaLoRA+RS on PFMs within both genome and language domain. For two genome understanding tasks, we observe that the OPTs achieve comparable performances with DNA-based PFMs.

2 Methods

In this section, we show the BBPE tokenization for the DNA sequences, the importance score computation, and the cubic budget schedule with random sampling.

BBPE tokenization for DNA sequences
In Table 1, we present three popular tokenizers for DNA sequences, labeled as (1), (2), and (3). The "words" tokenizer employs a dictionary derived from the four nucleotides. The tokenized sequence length of an input DNA equates to the number of nucleotides. However, this method lacks contextual information.

Table 1: Different tokenization algorithms for DNA sequences.

Original	AACTCAACGATC
(1) "words"	A A C T C A A C G A T C
(2) 6-mer	AACTCA ACTCAA CTCAAC TCAACG CAACGA AACGAT ACGATC
(3) BBPE	① 41 41 43 54 43 41 41 43 47 41 54 43 → A A C T C A A C G A T C
	② 4141 43 54 43 4141 43 47 41 54 43 → AA C T C AA C G A T C
	③ 414143 54 43 414143 47 41 54 43 → AAC T C AAC G A T C
	④ 414143 5443 414143 47 41 5443 → AAC TC AAC G A TC
	⑤ 414143 5443 414143 4741 5443 → AAC TC AAC GA TC

In contrast, the 6-mer tokenizer is gaining popularity in DNA sequencing (Dotan et al., 2023). The concept of k-mer revolves around extracting continuous subsequences of k nucleotides from a DNA sequence. However, one drawback of k-mer tokenization is the increased computational complexity, especially with larger k values. To address this, the BBPE tokenizer initializes a dictionary consisting of all individual bytes in UTF-8 encoding. It progressively selects the most frequent pairs of tokens to merge. Each combined pair is then added to the dictionary as a new token (shown in $① - ⑤$). OPTs are tailored with GPT-2's BBPE tokenizer (Radford et al., 2019). This tokenizer is well-suited for DNA sequences because it efficiently captures the recurring patterns of nucleotides. By focusing on the frequency of specific sequences, it offers a nuanced encoding that can illuminate biological motifs. This dictionary only consists of three tokens: "AAC", "TC", and "GA".

Importance Score PLMs contain many weight matrices to perform matrix multiplication. These weight matrices typically have full-rank. However, performing FMFT is not efficient. Thus, our goal is to reduce the number of ranks to project the high dimensional weights matrices to smaller subspaces. Mathematically, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d_p \times d_q}$, we approximate the gradient updates using singular value decomposition (SVD) for a low-rank representation, i.e., $W_0 + \Delta W = W_0 + P\Lambda Q$, where $P \in \mathbb{R}^{d_p \times r}$, $Q \in \mathbb{R}^{r \times d_q}$, and the rank $r \ll \min(d_p, d_q)$. Thus, for n matrices in the PLM, we need to perform SVD: $\Delta W_k = P_k \Lambda_k Q_k$ for $k = 1, ..., n$. Our importance score computation considers the importance from both singular values, which capture the magnitude of changes and indicate dominant variation directions, and singular vectors, which denote their orientations. Thus, each triplet $\Sigma_i = \{\lambda_{k,i}, P_{k,*i}, Q_{k,i*}\}$ is constructed with the i-th singular value $\lambda_{k,i}$ and the corresponding singular vectors $P_{k,*i}, Q_{k,i*}$. The importance score for each singular value is then computed as (Zhang et al., 2022): $S_{k_i} = s(\lambda_{k,i}) + \frac{1}{d_p}\sum_{j=1}^{d_p} s(P_{k,ji}) + \frac{1}{d_q}\sum_{j=1}^{d_q} s(Q_{k,ij})$. At each time step t, each entry in the matrix is associated with an importance score, computed as the product of its sensitivity and uncertainty, i.e., $s^t(w_{ij}) = \bar{I}^t(w_{ij})\bar{U}^t(w_{ij})$, where $\bar{I}^t(w_{ij})$ denotes the stabilized sensitivity and $\bar{U}^t(w_{ij})$ represents the stabilized uncertainty. These stabilized scores refine the original scores through a weighted adjustment. The updating rules for $\bar{I}^t(w_{ij})$ and $\bar{U}^t(w_{ij})$ are: $\bar{I}^t(w_{ij}) = \beta_1 \bar{I}^{t-1}(w_{ij}) + (1-\beta_1) I^t(w_{ij})$ and $\bar{U}^t(w_{ij}) = \beta_2 \bar{U}^{t-1}(w_{ij}) + (1-\beta_2)|I^t(w_{ij}) - \bar{I}^{t-1}(w_{ij})|$, where $I^t(w_{ij}) = |w_{ij} \bigtriangledown_{w_{ij}} \mathcal{L}^t|$ and $\mathcal{L}^t$ denotes the binary cross-entropy for a batch of data. Therefore, the sensitivity captures how much the loss responds to changes in a specific weight within a training batch. In contrast, uncertainty quantifies the fluctuations in the loss, given by $U^t(w_{ij}) = |I^t(w_{ij}) - \bar{I}^{t-1}(w_{ij})|$. The importance score is computed by balancing the two factors via $\bar{I}^t(w_{ij})\bar{U}^t(w_{ij})$.

Cubic Budget Schedule with Random Sampling We introduce a global budget, b_t, which diminishes following a cubic budget schedule defined as $b^t = b^T + (b^0 - b^T)(1 - \frac{t}{T})^3$. For the random sampling process, we define masks $R^t_{k,ii}$ for pruning λ^t_k and re-introducing the pruned λ^t_k. These masks are random variables derived from a Bernoulli distribution with parameter p, $R^t_{k,ii} \sim \text{Bernoulli}(p)$. The singular values to be retained are updated based on the following updating rule:

$$\hat{\Lambda}^t_{k,ii} = \begin{cases} \Lambda^t_{k,ii} \cdot (1 - R^t_{k,ii}) & \text{if } S^t_{k,i} \text{ is in the top } b^t \text{ of } S^t, \\ \Lambda^t_{k,ii} \cdot R^t_{k,ii} & \text{otherwise.} \end{cases} \tag{1}$$

Note that we here denote λ^t_k as $\Lambda^t_{k,ii}$ to more conveniently assign masks based on their position index i. Only those singular values, $\hat{\Lambda}^t_{k,ii}$, that meet both the random sampling criteria and have their importance score $S^t_{k_i}$ is in the top b^t of all scores S^t are retained. By introducing randomness rather than strictly adhering to a deterministic cutoff, the process becomes more robust against potential suboptimal initial states and inaccuracies in importance scores.

3 Experiments

3.1 Dataset and Setting-Up

We evaluate the FMFT, LoRA, AdaLoRA, and AdaLoRA+RS on PFMs using two genome understanding tasks, i.e., EMP on yeast and PD on humans. Please find the subsection 5.1 in the supplementary material for details.

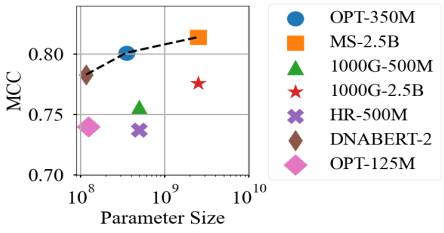

Figure 1: Parameter Size vs. MCC with Pareto front for FMFT.

3.2 Experimental Results

FMFT of OPTs performs on-par with DNA-centric PFMs on the H3 EMP task. The results for FMFT on different models are shown in Figure 1. We display the Matthews correlation coefficient (MCC) (Chicco and Jurman, 2020) for OPT-125M, OPT-350M, DNABERT-2, and four PFMs from NT, with sizes spanning from 500M to 2.5B parameters. These NT models have been pre-trained on three distinct datasets: the human reference genome (HR), the 1000G dataset, and genomes from multiple species (MS); therefore they are referred to as HR-500M, 1000G-500M, 1000G-2.5B, and

MS-2.5B. On FMFT of H3 EMP task, we find that the OPT with 350M paramters (OPT-350M) lies on the Pareto front (shown in black dotted line) with two DNA foundation models, i.e., DNABERT-2 and MS-2.5B. Specifically, the OPT-350M achieves a 2.5% decrease in MCC while utilizing a parameter size of 14% compared to MS-2.5B. The OPT model with 125M parameters (OPT-125M) outperforms the DNA foundation model HR-500M with 25% of HR-500M parameters. Despite OPTs being trained on natural languages, we find their FMFT performance highly competitive with DNA-centric PFMs, likely due to their strong reasoning ability (Coskun et al., 2023; Hu et al., 2023).

Benchmarking PEFT on DNA-centric PFMs and OPTs. In Table 2 and Table S2 (in subsection 5.2 of the supplementary material), we show the AUCs and MCCs for various models and methods on the PD and EMP tasks, respectively. The best results for each PFM are highlighted in bold, while the second best are in italic. In Table 2, AdaLoRA+RS consistently outperforms other PEFT techniques when applied to OPTs. Specifically, OPT-125M with AdaLoRA+RS achieves an AUC of 0.959 for the prom_all dataset. In comparison, AdaLoRA and LoRA attain AUCs of 0.902 and 0.887, respectively. Regarding DNA-focused models like 1000G-500M, AdaLoRA+RS's performance is comparable to that of FMFT. As an example, AdaLoRA+RS posts a 0.926 AUC with merely 6.9M trainable parameters (only 1.3% of FMFT's parameters), whereas FMFT achieves a 0.95 AUC using 500M trainable parameters. Furthermore, as evidenced by Table S2, AdaLoRA+RS's performance parallels that of FMFT.

Table 2: AUCs for various models and methods on the PD task.

Model	Method	# Train. Params.	Prom_all	Prom_notata	Prom_tata
DNABERT-2	FMFT	117M	0.908	0.950	0.804
	LoRA	1.6M	*0.918*	**0.971**	*0.812*
	AdaLoRA	1.0M	0.912	0.954	0.802
	AdaLoRA+RS	1.0M	**0.920**	*0.964*	**0.815**
1000G-500M	FMFT	500M	**0.950**	**0.951**	**0.939**
	LoRA	7M	0.921	0.942	0.899
	AdaLoRA	6.9M	0.924	0.949	0.871
	AdaLoRA+RS	6.9M	*0.926*	**0.951**	*0.918*
OPT-125M	FMFT	125M	0.898	*0.947*	0.864
	LoRA	1.1M	0.887	0.907	0.853
	AdaLoRA	1.0M	*0.902*	0.931	*0.886*
	AdaLoRA+RS	1.0M	**0.959**	**0.962**	**0.928**
OPT-350M	FMFT	350M	0.894	0.923	0.866
	LoRA	6.3M	0.917	0.928	0.904
	AdaLoRA	6.2M	*0.922*	*0.947*	*0.911*
	AdaLoRA+RS	6.2M	**0.938**	**0.956**	**0.929**

OPTs are strong genomic sequence learners. We also highlight the performances of OPTs in Table 3. This Table demonstrates the numbers of the performances for the PEFT method using OPTs are among the Top-2 for PD and EMP tasks, respectively. For the three datasets in PD task, OPT-125M with AdaLoRA+RS is among the Top-2 performed models in 3/3 tasks. Similaly, for the ten datasets in EMP task, OPT-350M with AdaLoRA+RS is among the Top-2 performed models in 4/10 tasks. This shows that when fine-tuning OPTs using genomic datasets with their BBPE encoding, they not only match the prowess of DNA-centric PFMs in the EMP task but also surpass the performance of fine-tuned DNA-centric PFMs in the PD task. This shows the adaptability of PLMs with BBPE encoding to genomic data, thereby illuminating the potential for a wider spectrum of applications for these natural language-based PLMs.

Table 3: # of performances for the method are among Top-2 for PD and EMP.

Model	Method	# Top-2 for PD	# Top-2 for EMP
OPT-125M	LoRA	0/3	2/10
	AdaLoRA	1/3	5/10
	AdaLoRA+RS	3/3	3/10
OPT-350M	LoRA	1/3	4/10
	AdaLoRA	2/3	5/10
	AdaLoRA+RS	2/3	4/10

4 Related Works and Conclusion

Recently, there have been significant advancements in the field of genomic domain, attributed to the development of DNA-centric Pre-trained Foundation Models (PFMs) such as DNABERT-2 (Zhou et al., 2023) and Nucleotide Transformer (NT) (Dalla-Torre et al., 2023). While Pre-trained Embedding Fine-Tuning (PEFT) is a well-established method in natural language-based Pre-trained Language Models (PLMs), there is a notable absence of mature applications of PLMs in DNA-centric PFMs. DNABERT-2 employs low-rank adapters (Hu et al., 2021) with a fixed number of ranks to curtail the quantity of trainable parameters. However, deterministic PEFTs can yield suboptimal performance when the initial states are less than ideal. To solve this issue, we propose an AdaLoRA (Zhang et al., 2023) with random sampling (AdaLoRA+RS) to prune and stochastically reintroduce pruned singular vectors. Our empirical observations on two genome understanding tasks demonstrate that OPT-350M, when combined with AdaLoRA+RS, positions itself on the Pareto front in comparison to its full-model fine-tuning baseline, utilizing merely 0.94% of the trainable parameters. Interestingly, we also discerned that natural language-based PLMs, exemplified by

OPT-125M, surpass the performance of the DNA foundation model HR-500M, despite using only 25% of the parameters.

References

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13.

Coskun, B., Ocakoglu, G., Yetemen, M., and Kaygisiz, O. (2023). Can chatgpt, an artificial intelligence language model, provide accurate and high-quality patient information on prostate cancer? *Urology*.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Carranza, N. L., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., Sirelkhatim, H., Richard, G., et al. (2023). The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01.

Dotan, E., Jaschek, G., Pupko, T., and Belinkov, Y. (2023). Effect of tokenization on transformers for biological sequences. *bioRxiv*, pages 2023–08.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, Y., Ameer, I., Zuo, X., Peng, X., Zhou, Y., Li, Z., Li, Y., Li, J., Jiang, X., and Xu, H. (2023). Zero-shot clinical entity recognition using chatgpt. *arXiv preprint arXiv:2303.16416*.

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

Ma, X., Fang, G., and Wang, X. (2023). Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. (2023). Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Zhang, Q., Zuo, S., Liang, C., Bukharin, A., He, P., Chen, W., and Zhao, T. (2022). Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International Conference on Machine Learning*, pages 26809–26823. PMLR.

Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., and Liu, H. (2023). Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*.

5 Supplementary Material

5.1 Dataset and Setting-Up

We evaluate the AdaLoRA+RS on PFMs using two genome understanding tasks, i.e., EMP on yeast and PD on humans. The statistics for both datasets are shown in Table S0.

Table S0: Genome understanding tasks

Task	Num. Datasets	Num. Classes	Sequence Length
EMP	10	2	500
PD	3	2	300

For FMFT, LoRA, AdaLoRA, and AdaLoRA+RS, we use a batch size of 8, while the evaluation batch size is set to 16. The model is trained over 5 epochs on all datasets for both tasks. We employ the Adam optimizer, with a learning rate of $3e-5$. Additionally, we implement a warmup ratio of 0.1 followed by linear decay. The L2 regularization weight decay is set at $5e-3$. For LoRA, we set the rank =8. For AdaLoRA and AdaLoRA+RS, the additional hyper-parameters are shown in Table S1. In this Table, Avg. b_0 and Avg. b_T denote the average number of singular values in each matrix. $(T_{\text{total}}-T)/T_{\text{total}}$ denotes the final fine-tune ratio after pruning and reintroducing, and ΔT indicates the intervals at which pruning and reintroducing are performed. The additoinal hyper-parameters for AdaLoRA are shown in gray columns and the additoinal hyper-parameters for AdaLoRA+RS is shown in all columns including the random sampling ratio p.

5.2 MCCs for various models and methods on the EMP task.

The MCCs for various models and methods on the EMP task is shown in Table S2. As evidenced by Table S2, AdaLoRA+RS's performance lies on a Pareto front with that of FMFT.

Table S1: Additoinal hyper-parameters for AdaLoRA (in gray) and AdaLoRA+RS (all columns).

Task	**Dataset**	**Avg.** b_0	**Avg.** b_T	$(T_{\text{total}} - T)/T_{\text{total}}$	ΔT	β_1	β_2	**Pruned Matrices**	p
EMP	H3	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K4me1	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K4me2	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K4me3	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K9ac	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K14ac	12	6	0.25	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K36me3	12	6	0.30	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H3K79me3	12	6	0.30	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H4	12	6	0.30	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
	H4ac	12	6	0.30	100	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}$	0.05
PD	Prom_all	8	6	0.15	5000	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}, W_o$	0.1
	Prom_notata	8	6	0.15	5000	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}, W_o$	0.1
	Prom_tata	8	6	0.15	5000	0.85	0.85	$W_q, W_k, W_v, W_{f_1}, W_{f_2}, W_o$	0.1

Table S2: MCCs for various models and methods on the EMP task.

Model	Method	# Train. Params.	H3	H3K4me1	H3K4me2	H3K4me3	H3K9ac	H3K14ac	H3K36me3	H3K79me3	H4	H4ac
DNABERT-2	FMFT	117M	*0.783*	**0.505**	*0.311*	**0.363**	**0.556**	0.526	**0.569**	**0.674**	**0.807**	**0.504**
	LoRA	1.6M	**0.791**	*0.451*	**0.342**	0	*0.543*	*0.523*	*0.565*	0.605	0.799	*0.456*
	AdaLoRA	1.0M	0.508	0.334	0.136	0.199	0.431	0.394	0.461	0.578	0.738	0.270
	AdaLoRA+RS	1.0M	0.734	0.450	0.300	*0.214*	0.479	**0.531**	0.557	*0.614*	*0.801*	0.449
1000G-500M	FMFT	500M	**0.756**	**0.379**	*0.288*	**0.288**	**0.488**	**0.399**	**0.461**	**0.579**	**0.752**	*0.342*
	LoRA	7M	0.725	0.354	0.240	0.267	0.452	0.387	0.442	0.515	0.685	0.331
	AdaLoRA	6.9M	0.734	0.365	0.284	*0.281*	0.469	0.390	0.452	0.557	0.704	0.340
	AdaLoRA+RS	6.9M	*0.749*	*0.367*	**0.289**	0.280	*0.474*	*0.395*	*0.460*	*0.562*	*0.717*	**0.343**
OPT-125M	FMFT	125M	**0.740**	**0.406**	0.216	**0.305**	**0.439**	*0.522*	**0.474**	0.505	**0.750**	**0.460**
	LoRA	1.1M	0.505	0.327	0.252	0.274	*0.429*	0.510	0.429	0.503	0.725	0.349
	AdaLoRA	1.0M	0.562	0.339	*0.275*	0.266	0.416	0.523	0.431	*0.513*	0.734	0.404
	AdaLoRA+RS	1.0M	*0.571*	*0.345*	**0.281**	*0.275*	0.428	**0.531**	*0.453*	**0.527**	*0.736*	*0.415*
OPT-350M	FMFT	350M	**0.801**	**0.463**	**0.287**	**0.289**	**0.504**	**0.517**	**0.505**	**0.645**	**0.789**	**0.471**
	LoRA	6.3M	0.619	0.406	0.203	0.209	0.439	0.447	0.449	0.504	0.722	0.437
	AdaLoRA	6.2M	0.627	0.409	0.209	0.217	0.446	0.459	0.451	0.525	0.731	0.448
	AdaLoRA+RS	6.2M	*0.636*	*0.411*	*0.213*	*0.229*	*0.457*	*0.463*	*0.455*	*0.527*	*0.741*	*0.451*