frontiers | Frontiers in Microbiology

TYPE Original Research
PUBLISHED 05 July 2023
DOI 10.3389/fmicb.2023.1215609

OPEN ACCESS

EDITED BY
Bowen Li,
Newcastle University, United Kingdom

REVIEWED BY
Guoqiang Xu,
Jiangnan University, China
Qi Dai,
Zhejiang Sci-Tech University, China

*CORRESPONDENCE
Ranran Huang
✉ huangrr@sdu.edu.cn

†These authors have contributed equally to this work

RECEIVED 02 May 2023
ACCEPTED 19 June 2023
PUBLISHED 05 July 2023

CITATION
Yang W, Li D and Huang R (2023) EVMP: enhancing machine learning models for synthetic promoter strength prediction by Extended Vision Mutant Priority framework. *Front. Microbiol.* 14:1215609. doi: 10.3389/fmicb.2023.1215609

EVMP: enhancing machine learning models for synthetic promoter strength prediction by Extended Vision Mutant Priority framework

Weiqin Yang[1,2†], Dexin Li[1,2†] and Ranran Huang[1*]

[1]Institute of Marine Science and Technology, Shandong University, Qingdao, China, [2]School of Computer Science and Technology, Shandong University, Qingdao, China

Introduction: In metabolic engineering and synthetic biology applications, promoters with appropriate strengths are critical. However, it is time-consuming and laborious to annotate promoter strength by experiments. Nowadays, constructing mutation-based synthetic promoter libraries that span multiple orders of magnitude of promoter strength is receiving increasing attention. A number of machine learning (ML) methods are applied to synthetic promoter strength prediction, but existing models are limited by the excessive proximity between synthetic promoters.

Methods: In order to enhance ML models to better predict the synthetic promoter strength, we propose EVMP(Extended Vision Mutant Priority), a universal framework which utilize mutation information more effectively. In EVMP, synthetic promoters are equivalently transformed into base promoter and corresponding *k*-mer mutations, which are input into BaseEncoder and VarEncoder, respectively. EVMP also provides optional data augmentation, which generates multiple copies of the data by selecting different base promoters for the same synthetic promoter.

Results: In Trc synthetic promoter library, EVMP was applied to multiple ML models and the model effect was enhanced to varying extents, up to 61.30% (MAE), while the SOTA(state-of-the-art) record was improved by 15.25% (MAE) and 4.03% (R^2). Data augmentation based on multiple base promoters further improved the model performance by 17.95% (MAE) and 7.25% (R^2) compared with non-EVMP SOTA record.

Discussion: In further study, extended vision (or *k*-mer) is shown to be essential for EVMP. We also found that EVMP can alleviate the over-smoothing phenomenon, which may contributes to its effectiveness. Our work suggests that EVMP can highlight the mutation information of synthetic promoters and significantly improve the prediction accuracy of strength. The source code is publicly available on GitHub: https://github.com/Tiny-Snow/EVMP.

KEYWORDS

EVMP, machine learning, promoter strength prediction, deep learning, synthetic promoter mutation library

1. Introduction

Promoters are the fundamental components of transcriptional regulation and have a direct impact on gene expression. In metabolic engineering and synthetic biology applications, promoters with desired strengths are critical (Gao et al., 2021). However, it is challenging for them to meet the requirements of the logical design and optimization of metabolic pathway due to the insufficient number of well-characterized promoters. Therefore, a vast library with hundreds of promoters with better properties must be created and characterized (Tang et al., 2020).

The conventional methods for building promoter libraries rely on functional module combinations or random sequence mutation techniques. Methods based on random sequence mutations, such as error-prone PCR, are regarded as a straightforward and effective mutagenesis technique and have been successfully used to synthesize artificial promoters. Alper et al. (2005), for instance, mutagenized the bacteriophage PL-promoter using error-prone PCR methods. Finally, 22 promoters with an intensity distribution range of 196 times were selected from nearly 200 promoter mutants. Zhao et al. (2021) constructed and characterized a mutant library of Trc promoters (P_{trc}) using 83 rounds of mutation-construction-screening-characterization engineering cycles, and established a synthetic promoter library that consisted of 3,665 different variants, displaying an intensity range of more than two orders of magnitude. Despite the availability of experimental methods, obtaining a small number of useful promoters from a random library typically necessitates time-consuming and laborious screening, and given the enormous number of possible sequence combinations, the effective identification of these useful promoters is largely constrained by a relatively small mutation library. For example, a 20-nucleotide sequence space represents $4^{20} = 1.1 \times 10^{12}$ possibilities, outnumbering even the largest bacterial libraries and most robust high-throughput screens. Therefore, obtaining the desired promoter through mutation, modification, or screening of existing promoters is challenging. In order to effectively guide how to search for new promoters in the huge potential sequence space, new methodologies should be developed to explore the relationship between promoter sequence and function more thoroughly (Cazier and Blazeck, 2021).

Fortunately, prediction of biological problems has been shown to be amenable to machine learning (ML) techniques, as comprehensively reviewed by de Jongh et al. (2020). Several recent studies have applied ML for promoter strength prediction. By varying a 50-nt region in the 5′ UTR of the HIS3 promoter, Cuperus et al. (2017) generated 500,000 variants for use as their training dataset for a convolutional neural network (CNN). This allowed them to predictably improve the expression of promoters with both random and native 5′ UTRs. They also demonstrated the advantage of using synthetic libraries as they found motifs that enhance transcription that are absent from the yeast genome. Additionally, ML has also enabled the prediction of transcriptional outputs for constitutive, inducible, or synthetic promoters in *Saccharomyces cerevisiae*. For instance, training a CNN with diversified libraries over 10^5 in size, created by precisely altering the P_{GPD} constitutive promoter and a P_{ZEV}-based inducible system, allowed prediction of promoter strengths with an accuracy of 79% (McIsaac et al., 2014; Kotopka and Smolke, 2020). Similarly, de Boer et al. (2020) created what is perhaps the largest synthetic library ever made for ML promoter engineering by synthesizing over 100 million random 80-bp upstream activating sequences (UASs). This library was used as a training dataset for a TF-motif-based model that was able to correctly predict the expression level of 94% of random promoters. In an elegant combination of hybrid promoter engineering and ML in human cells, Wu et al. (2019) created 129-bp UASs that contained tandem, human TF binding sites that represented over 6,000 unique motifs taken from two TF databases, and then used the resulting dataset to train a generalized linear model with elastic net regularization (GLMNET) (Kheradpour and Kellis, 2014; Weirauch et al., 2014; Wu et al., 2019). The model enables us to successfully predict the differential expression of individual promoters across different cell lines, a difficult task for mammalian cell engineering.

Although the progress made in the aforementioned works is exciting, their methods are not without flaws. The key characteristic of the synthetic promoter dataset is that the synthetic promoters are too close to each other while their strengths exhibit a substantial degree of variation (Zhao et al., 2021). This results in the inability to differentiate the strengths of synthetic promoters through sequence homology, and commonly used language models such as LSTM (Long Short-term Memory) also perform poorly. Therefore, we have considered another simple approach, which is to highlight the mutation information.

In this paper, we propose a novel framework, EVMP (Extended Vision Mutant Priority), to extract important mutation features and enhance ML models. In EVMP, synthetic promoters are equivalently transformed into EVMP format data, including base promoter and *k*-mer mutations (Liu and Wu, 2021), which are input into BaseEncoder and VarEncoder, respectively. We evaluated the effectiveness of EVMP on a Trc synthetic promoter library constructed by Zhao et al. (2021) and found that EVMP models exhibited varying degrees of improvement over non-EVMP models. Specifically, LSTM was improved by 61.30% (MAE) and 43 times (R^2), Transformer by 34.17% (MAE) and 29.93% (R^2), and the other models had effect improvements ranging from 5.15 to 7.27% (MAE) and 2.88 to 8.90% (R^2). Compared to the state-of-the-art (SOTA) non-EVMP method (Zhao et al., 2021), which was included in our experiments, EVMP achieved superior results and improved the SOTA non-EVMP record by 15.25% (MAE) and 4.03% (R^2). EVMP offers optional data augmentation, which involves selecting multiple base promoters and generating multiple copies of data in the EVMP format. Data augmentation further enhanced the effectiveness of EVMP, which was 17.95% (MAE) and 7.25% (R^2) higher than that of the SOTA non-EVMP method. We conducted ablation experiments to demonstrate the critical role of the extended vision, or the *k*-mer in mutation representation, in EVMP framework. Additionally, we discussed the effectiveness of EVMP and found that it can alleviate the over-smoothing phenomenon that often occurs in synthetic promoter datasets. Our work suggests that EVMP can significantly improve the performance of ML models in predicting synthetic promoter strength, while its effectiveness is accompanied by reasonable interpretability. To the best of our knowledge, this

is the first research to integrate mutation features for predicting promoter strength.

2. Materials and methods

2.1. EVMP architecture

As illustrated in Figure 1A, EVMP framework takes EVMP format data as input and predicts promoter strength using BaseEncoder and VarEncoder as backbone networks. The pipeline for applying the EVMP framework to the synthetic promoter strength prediction task consists of two steps: generating EVMP format data from the synthetic promoter dataset and passing the data through the EVMP framework to obtain prediction results.

2.1.1. EVMP data processing

In the synthetic promoter dataset, a *base promoter* $S = (x_1, \cdots, x_n)$ is first selected, where each x is a base $\in \mathbb{R}^d$ and $d = 5$ (representing one-hot encoding of four bases A, T, C, G, and B for the blank). For any synthetic promoter $S' = (x'_1, \cdots, x'_n)$, multiple mutation sites are obtained by pairing S' with the base promoter S, and the k-mer subsequences centered at each mutation site in synthetic promoter S' are referred to as *k-mer mutations* $M(S, S')$. Finally, we obtain the EVMP format data consisting of base promoter S and k-mer mutations $M(S, S')$, which is equivalent to the original synthetic promoter S', denoted as $\langle S, M(S, S')\rangle$.

2.1.2. Data augmentation

Data augmentation is an optional step in the EVMP pipeline, which is primarily achieved through selecting multiple base promoters. When data augmentation is not applied, the base promoter is set to P_{trc} by default, and each synthetic promoter S' is mapped to a corresponding data point $\langle P_{\text{trc}}, M(P_{\text{trc}}, S')\rangle$ in the EVMP format. By introducing data augmentation, we can select multiple (e.g., 10 in this experiment) base promoters, denoted as $P_1, P_2, \cdots$, and generate multiple data points for the same synthetic promoter S'. Specifically, we can obtain $\langle P_1, M(P_1, S')\rangle, \langle P_2, M(P_2, S')\rangle, \cdots$, each corresponding to a different base promoter but the same synthetic promoter. Choosing a suitable base promoter is often challenging. Data augmentation provides the model with more choices and opportunities to select a better base promoter.

2.1.3. EVMP framework

Similar to common natural language processing (NLP) models, the BaseEncoder in the EVMP pipeline directly processes the base promoter. Specifically, the BaseEncoder takes the base promoter S as input and produces the corresponding base embedding e_{base} as the output.

$$e_{\text{base}} := \text{BaseEncoder}(S) \tag{1}$$

In contrast to typical NLP models that handle the entire synthetic promoter, the VarEncoder in the EVMP pipeline deals with k-mer mutations $M(S, S')$ between the base promoter S and the synthetic promoter S'. Specifically, the VarEncoder receives the k-mer mutations that are represented by the mutation representation module and produces the corresponding mutation embedding e_{var} as the output.

$$e_{\text{var}} := \text{VarEncoder}(M(S, S')) \tag{2}$$

In Section 2.2, we will introduce two different mutation representation paradigms, namely Vars+PE and Mask.

In the EVMP framework, BaseEncoder and VarEncoder are model-agnostic and may not necessarily be distinct. Our study offers several alternatives for BaseEncoder and VarEncoder, including LSTM (Hochreiter and Schmidhuber, 1997; Gers et al., 2000), Transformer (Vaswani et al., 2017), Random Forests(RF) (Breiman, 2001), GBDT (Friedman, 2001), XGBoost (Chen and Guestrin, 2016), and SVM (Boser et al., 1992; Cortes and Vapnik, 1995). For deep learning models, BaseEncoder and VarEncoder can be separate, taking inputs S and $M(S, S')$, respectively. For conventional machine learning models, BaseEncoder and VarEncoder must be the same model, and the inputs S and $M(S, S')$ are concatenated. It should be noted that the BaseEncoder of the baseline models only refers to the encoder of the models, which means the FFN output layer is excluded. . In addition, Appendix 1 in Supplementary material provides the necessary mathematical foundations for the aforementioned machine learning methods. For more details on model implementations, see Appendix 2 in Supplementary material.

2.1.4. Promoter strength prediction

The base embedding e_{base} and mutation embedding e_{var} produced by the BaseEncoder and VarEncoder, respectively, are concatenated and passed through a feed-forward network (FFN) for predicting promoter strength. The model is trained on a dataset $D = \{(x_i, y_i)|i = 1, \cdots, N\}$ using the objective function:

$$\min_{w} \sum_{i=1}^{N} |\hat{y}_i - y_i| + \lambda \|w\|^2 \tag{3}$$

where $\hat{y}_i$ is the prediction of promoter strength of the EVMP format input x_i, w is the weights of model, and λ is the regularization constant.

2.2. Mutation representation paradigms

Mutation representation is a critical component of EVMP. Different mutation representation methods may be appropriate for different models. Nevertheless, in general, all mutation representation methods must satisfy two fundamental requirements: (i) the *extended vision* of mutation, and (ii) the preservation of mutation *positional information*. In the scope of our discussion, all mutation representation methods are based on k-mer mutations, which aligns with the extended vision principle stated in requirement (i). Therefore, the key to achieving mutation representation lies in preserving positional information. In the following, we present two different paradigms for mutation representation, including Vars+PE (used by LSTM and Transformer) and Mask (used by conventional ML models).

10.3389/fmicb.2023.1215609

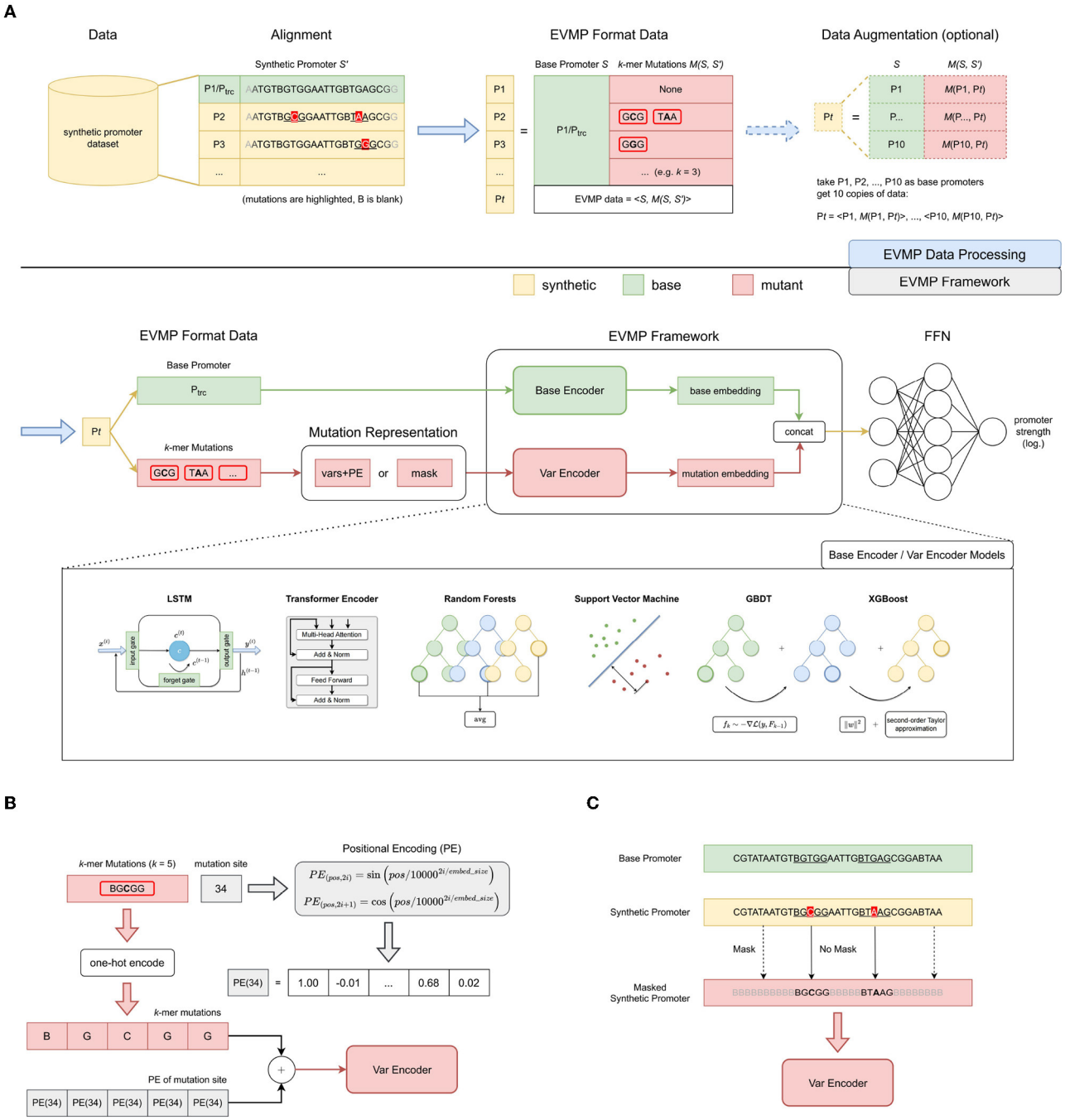

FIGURE 1
Extended Vision Mutant Priority framework. **(A)** The overall pipeline of EVMP framework for synthetic promoter strength prediction, which consists of data processing and framework. EVMP data processing is based on alignment, where the original synthetic promoter S' is equivalent to data in EVMP format: the base promoter S and the k-mer mutation $M(S, S')$. P_{trc} serves as the default base promoter, while data augmentation based on multiple base promoters is an optional step in data processing. EVMP framework consists of BaseEncoder and VarEncoder, which characterize the base promoter and the k-mer mutations after mutation representation, respectively. Base embedding and mutation embedding are then concatenated together and sent to a feed-forward network for strength prediction. **(B)** Vars+PE, one of the mutation representation paradigms. Each base in a k-mer mutation is added with the Positional Encoding of the mutation site. **(C)** Mask, one of the mutation representation paradigms. Each base that appears in a particular k-mer mutation is retained, and the rest is masked to B (blank).

2.2.1. Vars+PE

In Vars+PE, VarEncoder takes the sequence of k-mer mutations as input. However, using this form destroys the positional information of individual mutation sites. To incorporate positional information for mutations, one approach is to use Positional Encoding [PE, Vaswani et al. (2017)], which is a function $PE : \mathbb{R} \rightarrow \mathbb{R}^d$. Generally, the Positional Encoding is directly added to the sequence, and therefore, we refer to this

technique as *vars+PE*. In this study, we utilize the sin-cos Positional Encoding defined in Definition A1.4 in Appendix 1.4 in Supplementary material and Figure 1B.

$$M_{\text{Vars+PE}}(S, S') = \left\{ \left(x'_{i-\lfloor k/2 \rfloor} + PE(i), \cdots, x'_{i+k-\lfloor k/2 \rfloor - 1} + PE(i) \right) | \text{ if } x_i \neq x'_i, i = 1, \cdots, n \right\} \quad (4)$$

It should be noted that BaseEncoder and VarEncoder add PE in different ways. Specifically, BaseEncoder adds $PE(p)$ to the base at position p, bases at different positions are added different PE, while VarEncoder adds the same $PE(p')$ to k bases in a k-mer mutation at mutated position p', as shown in Equation (4).

2.2.2. Mask

In Section 2.2.1, we utilized the PE technique since the selection of k-mer mutations resulted in the disruption of the initial base order. However, if the promoter sequence is maintained in its original form, the provision of additional positional information is unnecessary. The *Mask* technique fits this need, which assigns bases to B(Blank) if they never appear in any of the k-mer mutations, so as to disguise the less significant bases.

$$M_{\text{Mask}}(S, S') = S' \cdot \left(\mathbb{I} \left(x_{i-k+\lfloor k/2 \rfloor + 1} \neq x'_{i-k+\lfloor k/2 \rfloor + 1} \vee \cdots \vee x_{i+\lfloor k/2 \rfloor} \neq x'_{i+\lfloor k/2 \rfloor} \right) | i = 1, \cdots, n \right) \quad (5)$$

As shown in Figure 1C, bases that appear in k-mer mutations are kept in their original positions, while the remaining bases are masked.

2.3. Datasets and models

The central experiment of this study is the prediction of synthetic promoter strength. The experiments are primarily based on the Trc synthetic promoter library created by Zhao et al. (2021), which consists of 3665 Trc synthetic promoters and uses log Fluorescence/OD600 as a measure of promoter strength. Previous research, such as that conducted by Zhao et al. (2021), has evaluated the performance of various models on this dataset, including LSTM, RF, XGBoost, and GBDT. Our non-EVMP baselines had included all these competitive models in previous work. In addition, models such as Transformer and SVM were added as options. For EVMP models, BaseEncoder and VarEncoder were the same, and their implementation referred to Section 2.1.3. EVMP is based on alignment, and we used MEGA (Kumar et al., 2008) to perform alignment between the base promoter and synthetic promoter.

Our main experiment used a 9:1 split of the dataset as training data and test data, and the training data was split into a 9:1 training set and validation set in each cross-validation. All experiments used 5-fold cross-validation.

An illustration of sequence homology analysis is necessary. In general promoter strength prediction, homologous sequences are considered likely to have similar strength, so homologous sequences in the training and test sets should be discarded. However, in mutation-based synthetic promoter libraries, all sequences are homologous, which renders the sequence homology approach ineffective. In Figure 2, the number of mutations vs. the strength difference between the pairwise promoters in the training and test sets that we divided is presented. As shown, although the average strength difference tends to decrease as the number of mutations decreases, the strength difference spans nearly three orders of magnitude regardless of the number of mutations. Therefore, it is considered inappropriate to suggest that promoters with less number of mutations and similar strength should be removed simply based on sequence homology—since there may be other promoters with a small number of mutations but very different strength. Therefore, we kept the original dataset to simulate the most realistic synthetic promoter library possible.

It should be noted that our results could not be directly compared to those in Zhao et al. (2021) for fairness since the proportion of the training set was reduced and different independent test sets were adopted. For a fair comparison, the competitive models in Zhao et al. (2021) were all retrained under the same conditions, and we adopted the source code of the original paper, located at https://github.com/YuDengLAB/Predictive-the-correlation-between-promoter-base-and-intensity-through-models-comparing. We used MAE (mean absolute error, primary) and R^2 (coefficient of determination, secondary) as evaluation metrics. In general, MAE is sensitive to the magnitude of promoter strength, while R^2 can reflect the overall prediction effect, both of which are meaningful evaluation metrics. The reason why MAE is used as the main metric is that we usually care more about high-strength promoters, thus it is more important to accurately predict the strength of synthetic promoters with larger values.

3. Results

3.1. EVMP discards redundant bases

In the synthetic promoter library, most of the bases between different promoters are identical, which limits the models' ability to gain informative features from these shared bases. As shown in Figure 3A, in comparison to P_{trc}, synthetic promoters with 2, 3, and 4 mutations account for the main part of the dataset, while the vast majority of synthetic promoters have no more than 8 mutations, which indicates that highlighting mutations is necessary.

In EVMP, most of the same bases are useless and should be discarded. We define the *receptive field* of VarEncoder as the proportion of the number of bases appearing in a k-mer mutation to the length of promoter sequence. As depicted in Figure 3B, when $k = 3$ and 8, the mean receptive fields of VarEncoder turn out to be 9 and 22, respectively, corresponding to 10.6 and 25.9% of the entire promoter length. The VarEncoder receives inputs that discard redundant bases while retaining necessary mutation information, making the EVMP approach distinct from traditional NLP models.

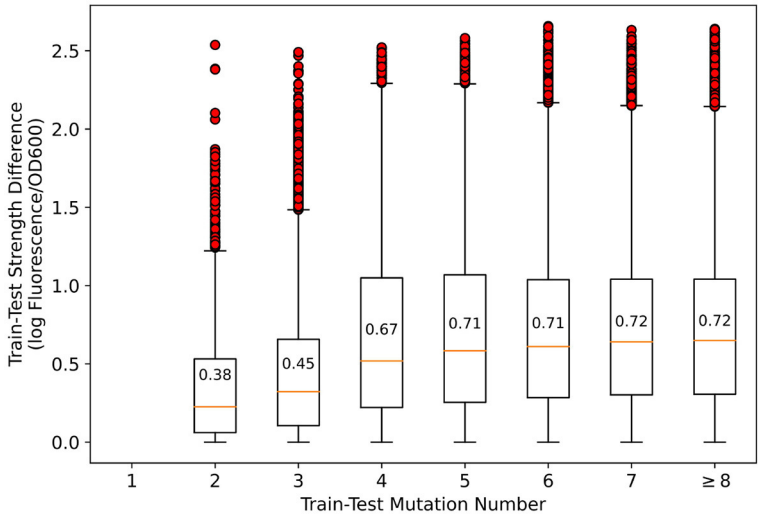

FIGURE 2
Relationship between the number of mutations and strength differences between training-test set pairwise promoters. In the figure, boxplots represent quartiles, and red circles represent outliers. The difference in promoter strength spanned nearly three orders of magnitude for all mutation numbers.

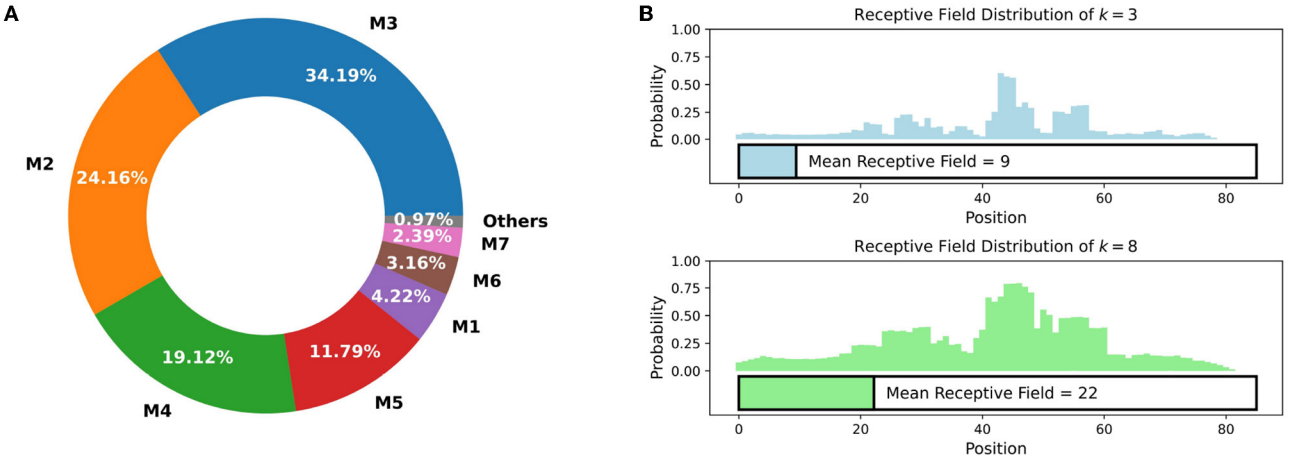

FIGURE 3
(A) The proportion of different mutation times in Trc synthetic promoter library. M*t* ($t = 1, 2, \cdots$) stands for synthetic promoters with *t* different mutation sites. The vast majority (> 99%) of synthetic promoters have no more than 8 mutation sites. **(B)** The probability that each position is included in the receptive field of VarEncoder, each promoter is filled **(B)** to 85 length. For $k = 3$ and 8, the mean receptive field of VarEncoder is 9 and 22, respectively, which is much less than the total length of the promoter.

3.2. EVMP enhances the effect of ML models

The results of the comparison between the effects of EVMP and non-EVMP models are presented in Table 1. Among the conventional machine learning models, the state-of-the-art (SOTA) non-EVMP model was RF (0.2262), while the MAE of the other non-EVMP models ranged from 0.23 to 0.26. However, the performance of non-EVMP LSTM (0.5337) and Transformer (0.2912) models was inferior to that of the other ML models.

After applying the EVMP method, MAEs of all models demonstrated varying degrees of improvement. The LSTM (0.5337 → 0.2065) and Transformer (0.2912 → 0.1917) showed a 61.30 and 34.17% improvement, respectively, surpassing the performance of the SOTA non-EVMP model (0.2262) and setting a new SOTA record. Other models also showed improvement ranging from 5.15 to 7.27%. The application of EVMP enhanced the performance of each model in predicting synthetic promoter strength, with the SOTA EVMP model (EVMP-Transformer, 0.1917) outperforming the SOTA non-EVMP model (RF, 0.2262) by 15.25%.

EVMP also made the R^2 of each model increase to varying degrees, and the R^2 improvement of the traditional machine learning model ranges from 2.88 to 8.90%. Transformer achieved 29.93% improvement in R^2. In particular, the R^2 of LSTM improves by a factor of 43—this is due to the fact that LSTM did not work at all before applying EVMP. EVMP also improved SOTA R^2 from

TABLE 1 Results of EVMP and non-EVMP models on synthetic promoter strength prediction task.

Method	Model	Mutant	k	Valid MAE	Test MAE	Enhance	SOTA
non-EVMP	SVM			0.2522 ± 0.0050	0.2604 ± 0.0012		0.2262
	GBDT*			0.2357 ± 0.0096	0.2434 ± 0.0042		
	XGBoost*			0.2193 ± 0.0107	0.2340 ± 0.0018		
	RF*			0.2273 ± 0.0098	0.2262 ± 0.0022		
	LSTM*			0.4959 ± 0.0213	0.5337 ± 0.0011		
	Transformer			0.2744 ± 0.0171	0.2912 ± 0.0020		
EVMP	SVM	Mask	8	0.2337 ± 0.0060	0.2414 ± 0.0012	7.27%	**0.1917**
	GBDT			0.2197 ± 0.0096	0.2285 ± 0.0030	6.14%	
	XGBoost			0.2051 ± 0.0143	0.2183 ± 0.0036	6.72%	
	RF			0.2231 ± 0.0132	0.2145 ± 0.0012	5.15%	
	LSTM	Vars+PE	3	0.1976 ± 0.0155	0.2065 ± 0.0065	**61.30%**	
	Transformer		5	**0.1839 ± 0.0126**	**0.1917 ± 0.0013**	**34.17%**	
				EVMP SOTA MAE Enhance			**15.25%**
Method	Model	Mutant	k	Valid R^2	Test R^2	Enhance	SOTA
non-EVMP	SVM			0.64 ± 0.05	0.63 ± 0.01		0.69
	GBDT*			0.68 ± 0.04	0.69 ± 0.01		
	XGBoost*			0.69 ± 0.05	0.69 ± 0.00		
	RF*			0.61 ± 0.03	0.65 ± 0.01		
	LSTM*			0.01 ± 0.01	0.02 ± 0.01		
	Transformer			0.55 ± 0.07	0.55 ± 0.01		
EVMP	SVM	Mask	8	0.66 ± 0.05	0.67 ± 0.00	5.68%	**0.72**
	GBDT			**0.70 ± 0.03**	**0.71 ± 0.01**	2.88%	
	XGBoost			**0.71 ± 0.05**	**0.72 ± 0.01**	4.34%	
	RF			0.65 ± 0.05	**0.71 ± 0.01**	8.90%	
	LSTM	Vars+PE	3	**0.70 ± 0.06**	**0.71 ± 0.02**	**4312.50%**	
	Transformer		5	**0.74 ± 0.05**	**0.71 ± 0.01**	**29.93%**	
				EVMP SOTA R^2 Enhance			**4.03%**

The entry *Mutant* is mutation representation method, k is the length of k-mer mutations, MAE is mean absolute error, R^2 is coefficient of determination, *Enhance* reflects the effect improvement of EVMP models compared to non-EVMP models, measured as the reduction in MAE and the increase of R^2. All MAEs and R^2s are the results of five-fold cross validation and are expressed as mean ± standard deviation. The model* annotated with an asterisk is derived from Zhao et al. (2021). Bold values are the best experimental results.

0.69 to 0.72. It can be seen that EVMP has a high improvement in both MAE and R^2, which indicates that EVMP can not only improve the overall prediction effect. An interesting finding is that R^2 improved less than MAE, indicating a clear rise in prediction accuracy for high-strength promoters.

3.3. Data augmentation further enhances EVMP

In our previous discussion, we did not incorporate optional data augmentation in the EVMP process. As described in Section 2.1.2, data augmentation involves selecting multiple base promoters $(P_1, P_2, \cdots, P_{10})$ and generating multiple copies of data $\langle P_1, M(P_1, S')\rangle$, $\langle P_2, M(P_2, S')\rangle$, $\cdots$, $\langle P_{10}, M(P_{10}, S')\rangle$ for each synthetic promoter S'. It is worth noting that the k-mer mutations $M(P_i, S')$ differ based on the selection of the base promoter $P_i, i = 1, \cdots, 10$. The ten selected base promoters, which differ from P_{trc} by 2 to 7 different sites, are listed in Table A4 in Appendix 2.

Table 2 presents the relevant comparative experiments of data augmentation. In these experiments, we chose EVMP-Transformer ($k = 5$) model for analysis. The first two Fixed-Fixed experiments were the same as in Table 1. The third Rand-Rand experiment randomly selected one of the 10 base promoters $P_1, \cdots, P_{10}$ as the base promoter, and each synthetic promoter may had a different base promoter. The fourth Augmented-Augmented experiment used all 10 base promoters and expanded the dataset by a factor of 10 by applying data augmentation.

Compared with the Fixed-Fixed method, the EVMP Rand-Rand method (0.2396, 0.59) exhibited weaker performance than the EVMP Fixed-Fixed method (0.1917, 0.71), but still outperformed

the non-EVMP Fixed-Fixed method (0.2912, 0.55). It is important to note that the Rand-Rand method represented an extreme scenario, where only approximately 1/10 of the original dataset was utilized by directly using P_{trc} as the base promoter. Nevertheless, the Rand-Rand method still achieved better performance than the non-EVMP method, which highlights the robustness of EVMP.

The Augmented-Augmented method with data augmentation (EVMP+DA) achieved the best performance (0.1856, 0.74) among all experiments, which was 17.95% (MAE) and 7.25% (R^2) better than the non-EVMP SOTA. After data augmentation, multiple copies of data in the EVMP format were generated for each synthetic promoter from different perspectives (base promoters). In addition, the Augmented-Fixed method (not listed in Table 2), achieved a MAE of 0.1876 on the same Fixed (P_{trc}-based) test set in the main experiment, which was better than the EVMP method (0.1917, 0.71) and worse than the average effect of all base promoters (0.1856, 0.74). These results suggest that data augmentation indeed increased the diversity of the data to improve the model's performance, and was able to identify a better base promoter than P_{trc}. Therefore, data augmentation is a simple and effective alternative to rationally selecting base promoters.

TABLE 2 Results of EVMP-Transformer ($k = 5$) in synthetic promoter strength prediction task with data augmentation.

EVMP	DA	DA method		Valid MAE	Test MAE
		Train	Test		
×	×	Fixed	Fixed	0.2744 ± 0.0171	0.2912 ± 0.0020
✓	×	Fixed	Fixed	0.1839 ± 0.0126	0.1917 ± 0.0013
✓	×	Rand	Rand	0.2423 ± 0.0076	0.2396 ± 0.0142
✓	✓	Augmented	Augmented	**0.1824 ± 0.0176**	**0.1856 ± 0.0008**
		EVMP+DA SOTA MAE Enhance			**17.95%**
EVMP	DA	DA Method		Valid R^2	Test R^2
		Train	Test		
×	×	Fixed	Fixed	0.55 ± 0.07	0.55 ± 0.01
✓	×	Fixed	Fixed	0.74 ± 0.05	0.71 ± 0.01
✓	×	Rand	Rand	0.62 ± 0.03	0.59 ± 0.04
✓	✓	Augmented	Augmented	**0.76 ± 0.06**	**0.74 ± 0.05**
		EVMP+DA SOTA R^2 Enhance			**7.25%**

The entries *EVMP* and *DA* indicate whether to use EVMP and whether to use data augmentation, respectively. The entries *Train/Valid* and *Test* refer to the dataset processing method, where *Fixed* refers to using P_{trc} as base promoter, *Rand* refers to randomly selecting one of the 10 base promoters $P_1, \cdots, P_{10}$ as base promoter, and *Augmented* refers to generating 10 copies of data by using all 10 base promoters. All MAEs and R^2 are the results of cross validation. Bold values are the best experimental results.

3.4. Extended vision is necessary for EVMP

Section 2.2 emphasized the crucial importance of extended vision for the effectiveness of EVMP. The inclusion of k-mer mutations with $k > 1$ achieves this extended vision. Although it may seem that only including single base mutations ($k = 1$) in the mutation input is the most straightforward approach, a single base may not provide sufficient information for the promoter. As a result, many biological software programs, such as Allesøe et al. (2021) and Nurk et al. (2017), use k-mers. Experimental results in Figures 4A, B, which show the Test MAE of EVMP-Transformer and EVMP-RF for different values of $k = 1, 3, 5, 8$, support this claim.

The Test MAE and R^2 of EVMP-Transformer with $k = 5, 8$ was the best with values of (0.1917, 0.66) and (0.1952, 0.71), compared to (0.3136, 0.47) for $k = 1$, (0.1998, 0.70) for $k = 3$. The Test MAE and R^2 of EVMP-RF with $k = 8$ was the best with a value of (0.2160, 0.71), compared to (0.2566, 0.58) for $k = 1$, (0.2212, 0.66) for $k = 3$, (0.2221, 0.66) for $k = 5$. These results indicate that $k > 1$ is much better than $k = 1$ and that extended vision is an indispensable key to the success of EVMP. Furthermore, the effect of the models did not change significantly as k increased. Therefore, choosing a suitable k is not difficult, and the key is to ensure that $k > 1$. To avoid excessive overlap between k-mer mutations, we suggest using the formula $k = l/\alpha$, where l is the length of the promoter and α is the average number of mutation sites.

4. Discussion

4.1. Why EVMP: experimental perspective

Experimental results have shown that EVMP can significantly enhance the effect of ML models in predicting the strength of synthetic promoters. However, why EVMP is effective remains to be further explored. We found that EVMP reduces the *over-smoothing* phenomenon, which may contributes to its effectiveness. Over-smoothing phenomenon occurs when the outputs of different inputs appear to be almost the same. Appropriate smoothing is regarded as a symbol of robustness in other tasks like image recognition (Ruderman et al., 2018). However, considering the fact that the differences between synthetic promoters are much shallower, over-smoothing phenomenon is likely to lead to a poor performance.

We define the *average embedding distance* to study over-smoothing phenomenon. Assume that synthetic promoters are $P_1, \cdots, P_N$, and P_0 is the base promoter (P_{trc}). The output embedding vector $e_l(P) \in \mathbb{R}^{n_l}$ are calculated for each layer l and each input P. Denote the average embedding distance d_l of each layer as follows:

$$d_l = \frac{1}{N} \sum_{i=1}^{N} \frac{\|(e_l(P_i) - e_l(P_0)\|_2}{\sqrt{n_l}} \tag{6}$$

Average embedding distance measures the average distance of each dimension of the embedding at the l-th layer, thus it is capable to represent the dispersion of the output of each layer. In particular, a higher average embedding distance indicates a lighter over-smoothing phenomenon.

In Figure 5, the average embedding distance of each layer was calculated for EVMP and non-EVMP implementations of LSTM and Transformer, respectively, where LSTM calculated the average embedding distance of each time step. Figure 5B showed that the average embedding distance of EVMP-Transformer was always much higher than that of non-EVMP Transformer. Figure 5C showed that the growth rate of the average embedding

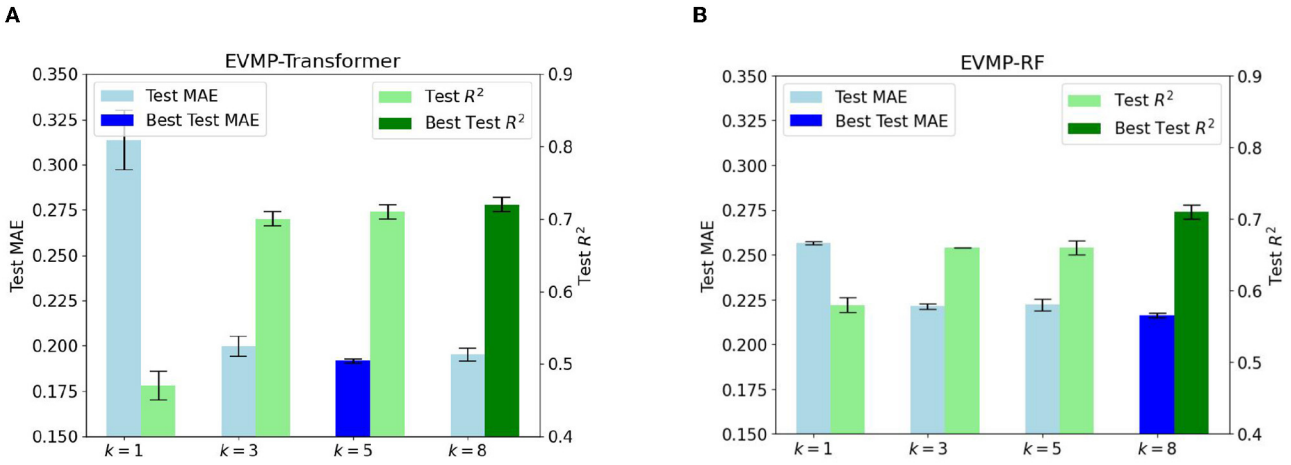

FIGURE 4
Test MAE and R^2 of EVMP models with different $k = 1, 3, 5, 8$ (length of k-mer Mutations). As k increased, the effect of models first improved and then tended to be the same. All MAEs and R^2s are the results of cross validation. **(A)** Test MAE and R^2 diagrams of EVMP-Transformer $k = 5, 8$ obtained the best result. **(B)** Test MAE and R^2 diagrams of EVMP-RF, $k = 8$ obtained the best result.

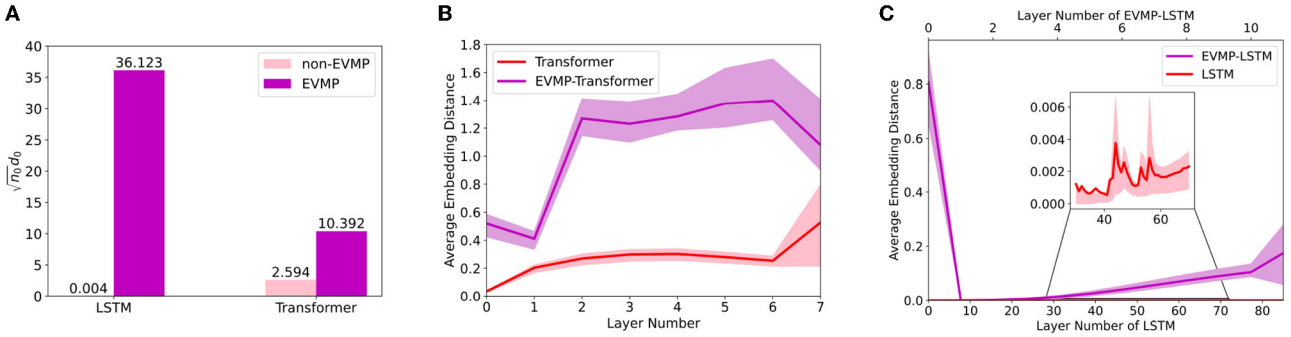

FIGURE 5
Average embedding distance for each layer of deep learning models, with the 0-th layer serving as the input. Since BaseEncoder receives the same base promoter P_{trc}, non-EVMP models calculate the average embedding distance of each layer directly, while EVMP models calculate the average embedding distance of VarEncoder. In synthetic promoter strength prediction, a larger average embedding distance is considered as the model is better able to discriminate subtle differences between synthetic promoters. **(A)** Non-norm d_0 ($\sqrt{n_0}d_0$) for LSTM and Transformer, which reflects the degree of dispersion of the input data. **(B)** Average embedding distance for each layer of Transformer and EVMP-Transformer. **(C)** Average embedding distance for each layer of LSTM and EVMP-LSTM, where LSTM calculates the average embedding distance for each time step.

distance of EVMP-LSTM from 0-th time step was much higher than that of non-EVMP LSTM. The above experimental results demonstrate that EVMP can effectively alleviate the over-smoothing phenomenon.

4.2. Why EVMP: theoretical perspective

Section 4.1 offers a comparison of the average embedding distance among layers of neural networks. The purpose is to verify intuitively whether EVMP mitigates the over-smoothing phenomenon and to elucidate the effectiveness of EVMP. In this context, a theoretical illustration is presented to support the validity of this perspective.

Zou et al. (2020) proved that when the width of each layer in deep ReLU networks is at least $\widetilde{\Omega}\left(n^{14}L^{16}/\phi^4\right)$, gradient descent can achieve zero training error within $\widetilde{O}\left(n^5L^3/\phi\right)$ iterations, where n is the number of training examples, L is the number of hidden layers, and ϕ is the maximum lower bound of the L_2-norm between every pair of the training samples.

Here, we regard $\sqrt{n_0}d_0$ as an approximation of ϕ, where n_0 is the dimension of input. As shown in Figure 5A, EVMP resulted in a noticeable increase in ϕ for deep learning models. Since the size of the training set n and the number of model layers L are remained unchanged, the overparameterizations of layer width and training iterations are reduced, and the model fitting effect tends to be improved. This statement is also enabled to be applied to RNN and CNN (Zou et al., 2020). The above discussion shows that EVMP can indeed alleviate the over-smoothing phenomenon, which contributes to its effectiveness.

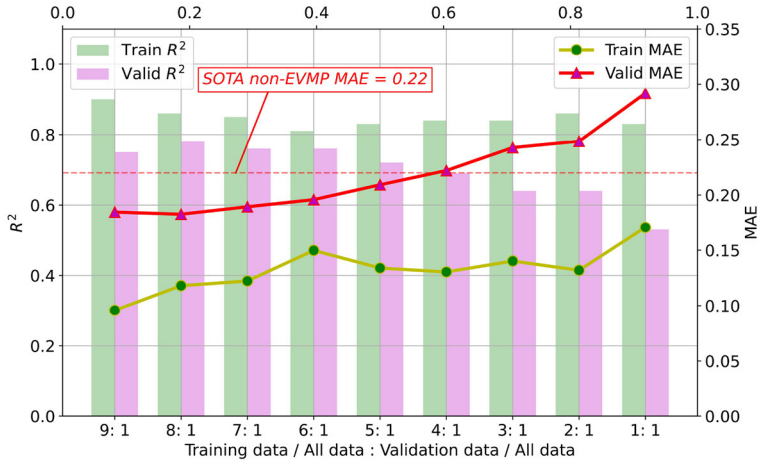

FIGURE 6
As the proportion of the training set decreased, the MAE of the validation set increased and the R^2 decreased. When the proportion of training set to validation set is higher than 5:1, EVMP-Transformer achieved lower MAE than SOTA non-EVMP models.

4.3. EVMP reduces the need for annotated data under the same prediction accuracy

Enhancing performance of ML models is just one aspect of EVMP. Another contribution of EVMP is to reduce the requirement for costly strength-annotated synthetic promoters under the original prediction accuracy. Here, given a fixed validation set proportion (1/10 of the original dataset), we randomly selected 1/9, 2/9, ⋯, 9/9 of the remaining 9/10 dataset as training set proportion, respectively. Figure 6 shows MAE and R^2 (coefficient of determination) of training and validation sets for each EVMP-Transformer model.

According to our experimental results in Table 1, the SOTA non-EVMP MAE was approximately 0.22, while Figure 6 shows that EVMP only needed a training set of 5/9 original size to achieve this effect. In other words, EVMP only needed 56% of the original data while maintaining the same prediction accuracy, which greatly reduces the dependence on high-quality labeled data.

4.4. Limitations and potential

EVMP has some limitations, that is, EVMP is only applicable to synthetic promoter libraries based on random mutations. For synthetic libraries built with other methods, EVMP does not necessarily yield gains, but it is no worse, since we can always build the original method as the BaseEncoder.

However, EVMP also has great potential. On the one hand, random mutation is still the most convenient method to construct synthetic libraries without prior knowledge. On the other hand, EVMP is not only suitable for synthetic promoters, but also theoretically applicable to other kinds of synthetic biological sequences (e.g., protein). In addition, in the discussion of data augmentation, we actually recognized that EVMP has enough capacity to handle multiple different synthetic promoter libraries, thus it has high scalability.

EVMP is a highly conceptual framework, and it is convenient to add additional components to EVMP. For example, if we want to use EVMP for protein downstream tasks, then additional information beyond sequence such as 3D structure information is a worthwhile feature to include, and multiple corresponding encoders can also work in parallel. The high robustness and scalability of EVMP provide the potential for it to be useful in more bioinformatics tasks.

The mutation modeling of EVMP is also worthy of further study, which includes 3D structural and different granularity alignments, the development of mutation representations other than Vars+PE and Mask, etc.

Finally, a suitable information interaction or aggregation between the encoders may also be required. With the development of attention technique, proper attention between different features and their embeddings is regarded as a suitable information interaction method.

5. Conclusions and future works

In this work, we proposed EVMP as a universal framework to enhance machine learning models for synthetic promoter strength prediction, which includes two steps: EVMP data processing and EVMP framework. The original synthetic promoter data are transformed into base promoter and *k*-mer mutations, which are processed by BaseEncoder and VarEncoder, respectively, to highlight mutation information. EVMP enhanced many models and achieved new SOTA records. EVMP also provides optional data augmentation based on multiple base promoters, which can further improve the performance of EVMP. In further study, We experimentally verified that extended vision, or

k-mer, is critical for the effectiveness of EVMP. In terms of interpretability, EVMP is proved to be able to alleviate the over-smoothing phenomenon and thus improves the effect of models.

EVMP is a highly robust and versatile machine learning enhancement framework with the potential to be extended to various mutation-based synthetic biology component libraries. In future research, more features and components, information interaction between encoders, and the application of EVMP in other synthetic biological sequence tasks are worthy of further exploration.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

WY and DL conceived and designed the study. WY conducted the experiments and analyzed the results of the experiment. WY, DL, and RH contributed to the writing of the manuscript. DL and RH contributed to the manuscript editing. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

Thanks for the support of the National Natural Science Foundation of China (Grant Numbers 31970113 and 32170065).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmicb.2023.1215609/full#supplementary-material

References

Allesøe, R. L., Lemvigh, C. K., Phan, M. V., Clausen, P. T., Florensa, A. F., Koopmans, M. P., et al. (2021). Automated download and clean-up of family-specific databases for kmer-based virus identification. *Bioinformatics* 37, 705–710. doi: 10.1093/bioinformatics/btaa857

Alper, H., Fischer, C., Nevoigt, E., and Stephanopoulos, G. (2005). Tuning genetic control through promoter engineering. *Proc. Natl. Acad. Sci. U.S.A.* 102, 12678–12683. doi: 10.1073/pnas.0504604102

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152. doi: 10.1145/130385.130401

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Cazier, A. P., and Blazeck, J. (2021). Advances in promoter engineering: novel applications and predefined transcriptional control. *Biotechnol. J.* 16, 2100239. doi: 10.1002/biot.202100239

Chen, T., and Guestrin, C. (2016). "XGboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. doi: 10.1145/2939672.2939785

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297.

Cuperus, J. T., Groves, B., Kuchina, A., Rosenberg, A. B., Jojic, N., Fields, S., et al. (2017). Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences. *Genome Res.* 27, 2015–2024. doi: 10.1101/gr.224964.117

de Boer, C. G., Vaishnav, E. D., Sadeh, R., Abeyta, E. L., Friedman, N., and Regev, A. (2020). Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nat. Biotechnol.* 38, 56–65. doi: 10.1038/s41587-019-0315-8

de Jongh, R. P., van Dijk, A. D., Julsing, M. K., Schaap, P. J., and de Ridder, D. (2020). Designing eukaryotic gene expression regulation using machine learning. *Trends Biotechnol.* 38, 191–201. doi: 10.1016/j.tibtech.2019.07.007

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 29, 1189–1232. doi: 10.1214/aos/1013203451

Gao, J., Jiang, L., and Lian, J. (2021). Development of synthetic biology tools to engineer pichia pastoris as a chassis for the production of natural products. *Synth. Syst. Biotechnol.* 6, 110–119. doi: 10.1016/j.synbio.2021.04.005

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: continual prediction with LSTM. *Neural Comput.* 12, 2451–2471. doi: 10.1162/089976600300015015

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.

Kheradpour, P., and Kellis, M. (2014). Systematic discovery and characterization of regulatory motifs in encode tf binding experiments. *Nucleic Acids Res.* 42, 2976–2987. doi: 10.1093/nar/gkt1249

Kotopka, B. J., and Smolke, C. D. (2020). Model-driven generation of artificial yeast promoters. *Nat. Commun.* 11, 1–13. doi: 10.1038/s41467-020-15977-4

Kumar, S., Nei, M., Dudley, J., and Tamura, K. (2008). MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. *Brief. Bioinform.* 9, 299–306. doi: 10.1093/bib/bbn017

Liu, W.-L., and Wu, Q.-B. (2021). Analysis method and algorithm design of biological sequence problem based on generalized k-mer vector. *Appl. Math. A J. Chin. Univ.* 36, 114–127. doi: 10.1007/s11766-021-4033-x

McIsaac, R. S., Gibney, P. A., Chandran, S. S., Benjamin, K. R., and Botstein, D. (2014). Synthetic biology tools for programming gene expression without nutritional perturbations in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* 42, e48. doi: 10.1093/nar/gkt1402

Nurk, S., Meleshko, D., Korobeynikov, A., and Pevzner, P. A. (2017). metaSPAdes: a new versatile metagenomic assembler. *Genome Res.* 27, 824–834. doi: 10.1101/gr.213959.116

Ruderman, A., Rabinowitz, N. C., Morcos, A. S., and Zoran, D. (2018). Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. *arXiv preprint arXiv:1804.04438*. doi: 10.48550/arXiv.1804.04438

Tang, H., Wu, Y., Deng, J., Chen, N., Zheng, Z., Wei, Y., et al. (2020). Promoter architecture and promoter engineering in *Saccharomyces cerevisiae*. *Metabolites* 10, 320. doi: 10.3390/metabo10080320

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. doi: 10.5555/3295222.3295349

Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., et al. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell* 158, 1431–1443. doi: 10.1016/j.cell.2014.08.009

Wu, M.-R., Nissim, L., Stupp, D., Pery, E., Binder-Nissim, A., Weisinger, K., et al. (2019). A high-throughput screening and computation platform for identifying synthetic promoters with enhanced cell-state specificity (specs). *Nat. Commun.* 10, 1–10. doi: 10.1038/s41467-019-10912-8

Zhao, M., Yuan, Z., Wu, L., Zhou, S., and Deng, Y. (2021). Precise prediction of promoter strength based on a *de novo* synthetic promoter library coupled with machine learning. *ACS Synth. Biol.* 11, 92–102. doi: 10.1021/acssynbio.1c00117

Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2020). Gradient descent optimizes over-parameterized deep ReLU networks. *Mach. Learn.* 109, 467–492. doi: 10.1007/s10994-019-05839-6