



OPEN ACCESS

EDITED BY

Gary Antonio Toranzos,
University of Puerto Rico, Puerto Rico

REVIEWED BY

Amjed Alsultan,
University of Al-Qadisiyah, Iraq
Yanmei Sun,
Northwest University, China

*CORRESPONDENCE

Minfeng Xiao
✉ xiaominfeng@genomics.cn
Min Fang
✉ fangmin@hit.edu.cn

RECEIVED 25 May 2025

ACCEPTED 28 August 2025

PUBLISHED 18 September 2025

CITATION

Chen Q, Zhao Z, Li M, Song W, Xiao M and Fang M (2025) MoEPH: an adaptive fusion-based LLM for predicting phage-host interactions in health informatics. *Front. Microbiol.* 16:1634705. doi: 10.3389/fmicb.2025.1634705

COPYRIGHT

© 2025 Chen, Zhao, Li, Song, Xiao and Fang. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](#). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

MoEPH: an adaptive fusion-based LLM for predicting phage-host interactions in health informatics

Qian Chen ¹, Zihang Zhao ², Min Li ^{3,4}, Wenchen Song ³,
Minfeng Xiao ^{3*} and Min Fang ^{5*}

¹School of Artificial Intelligence and National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, Guangdong, China, ²School of Computer Science and Technology, The University of Hong Kong, Hong Kong, Hong Kong SAR, China, ³BGI-Shenzhen, Shenzhen, China, ⁴University of Chinese Academy of Sciences, Beijing, China, ⁵Education Center of Experiments and Innovations, Harbin Institute of Technology, Shenzhen, Guangdong, China

Phage-host interaction prediction plays a crucial role in the development of phage therapy, particularly in combating antimicrobial resistance (AMR). Current in silico models often suffer from limited generalizability and low interpretability. To address these gaps, we introduce MoEPH (Mixture-of-Experts for Phage-Host prediction), a novel framework that integrates transformer-based protein embeddings (ProtBERT and Prot5) with domain-specific statistical descriptors. Our model dynamically combines features using a gated fusion mechanism, ensuring robust and adaptive prediction. We evaluate MoEPH on three publicly available phage-host interaction databases: Dataset 1 (101 host strains, 129 phages), Dataset 2 (38 host strains, 176 phages), and Dataset 3 (combined). Experimental results demonstrate that MoEPH outperforms existing methods, achieving an accuracy of 99.6% on balanced datasets and a 31% improvement on highly imbalanced data. The model provides a transparent, dynamic and knowledge-driven fusion solution for phage-host prediction, contributing to more effective phage therapy recommendations. Future work will focus on incorporating structural protein features and exploring alternative neural backbones for further enhancement.

KEYWORDS

trustworthy phage-host prediction, interpretable, robustness, transformer-based protein embeddings, mixture of experts (MoE)

1 Introduction

Antimicrobial resistance (AMR) has emerged as a global health crisis and a “silent pandemic,” threatening effective infection treatment worldwide. Without intervention, AMR is projected to cause up to 10 million deaths annually by 2050 (Walsh et al., 2023), with millions of lives already impacted each year. This dire situation has reignited interest in alternative therapeutics (Murray et al., 2022). Bacteriophage (phage) therapy—the use of viruses that specifically infect bacteria—has re-emerged as a promising strategy to combat drug-resistant infections. Phages can lyse antibiotic-resistant bacteria with high specificity, offering a potential lifeline where antibiotics fail. In this context, developing accurate phage–host identification methods is crucial to actualize phage therapy against AMR. Our work lies at the intersection of phage therapy and artificial intelligence, aiming to advance trustworthy medical AI to address this challenge. Moreover, increasing biological data resources facilitate this goal; for instance, we leverage a large phage–host interaction dataset

from BGI-Shenzhen,¹ which provides extensive phage genomic and host information to support model training and evaluation.

Existing *in silico* phage–host prediction methods, however, face significant limitations that hinder clinical utility. Early approaches rely on genomic sequence similarity or alignment-based heuristics, such as BLAST hits or CRISPR-spacer matches. Alignment-free statistical methods [e.g., WIsH (Galiez et al., 2017) and RaFAH (Coutinho et al., 2021)] predict hosts based on k-mer composition or protein content, but often fail on distantly related phages and have limited accuracy. Later methods introduced machine learning on genomic features: for example, logistic matrix factorization models (Leite et al., 2018) and network-based frameworks like VirHostMatcher-Net (Wang et al., 2020) integrated multiple genomic similarity measures to improve predictions. These offered moderate performance gains yet still struggle with generalizability and robustness, especially for novel phages or under-represented hosts. Deep learning has also been applied—notably PredPHI (Li et al., 2020), a CNN-based tool utilizing phage and host protein features—achieving higher accuracy than classical approaches. However, such deep models act as black boxes with limited interpretability, and their improvements in accuracy remain modest. Even knowledge-integrated approaches encounter challenges: a recent knowledge graph model KGVHI (Pan et al., 2024) combines multiple data types (genomic, proteomic, and host metadata) to predict microbe–host pairs and shows excellent performance on benchmark datasets, yet it requires comprehensive prior knowledge and is not specialized for bacteriophage therapy scenarios. In summary, existing methods tend to overfit to training data and imbalanced distributions, often over-predicting dominant hosts while missing rare interactions. They lack the adaptability to generalize to new phage or host species and provide little biological insight into predictions. Such opacity and instability undermine user trust, which is critical for AI deployment in healthcare. Thus, improvements in generalization, robustness, and explainability are essential before phage–host prediction models can be used confidently in clinical practice.

Meanwhile, large pre-trained models have brought transformative advances to bioinformatics. In the protein biology domain, protein language models (pLMs) like ProtBERT and ProT5 (Elnaggar et al., 2021) leverage Transformer architectures to learn rich protein sequence representations. These models capture structural motifs and achieve state-of-the-art performance in diverse biological tasks. Treating protein sequences as a “language of life,” such models offer superior feature learning for tasks like phage–host prediction. Similarly, in biomedical NLP, large domain-specific language models [e.g., PubMedBERT (Gu et al., 2021)] have demonstrated that pre-training on in-domain data yields powerful representations for downstream tasks. However, current pLM-based phage–host prediction approaches are often used only as static feature extractors, lacking dynamic adaptation or integration of domain knowledge. Recent studies have begun to endow language models with biological knowledge or multi-modal data (Chen et al., 2024), but a cohesive framework that combines

pre-trained embeddings with adaptive, knowledge-driven fusion for phage–host prediction remains absent.

To address these gaps, we propose MoEPH, a Mixture-of-Experts framework designed to improve both performance and transparency for trustworthy phage–host prediction. MoEPH employs multiple expert subnetworks specializing in different feature modalities, with a gating network dynamically selecting experts for each input (Shazeer et al., 2017). This architecture enables adaptive, context-specific learning, effectively capturing both genomic composition signals and high-level protein patterns. By synergistically integrating pre-trained protein embeddings with interpretable statistical features, MoEPH achieves not only superior predictive performance but also enhanced explainability. Specifically, our model maintains robust accuracy even on highly imbalanced and novel data, mitigating the overfitting to dominant hosts that plagues prior methods. It also provides interpretability through per-sample expert weight analysis, offering biological insight into which features drive a given prediction. These qualities align with key pillars of trustworthy AI—reliability, explainability, and adaptability (Aljohani et al., 2025)—making MoEPH particularly suitable for sensitive applications like phage therapy recommendation. In summary, our main contributions are an innovative multi-expert fusion strategy tailored for phage–host prediction, a comprehensive evaluation demonstrating state-of-the-art performance across multiple datasets, and an analysis showing improved generalization and interpretability compared to existing models.

In summary, our main contributions are:

- **Trusted phage–host prediction framework:** We develop MoEPH, combining pre-trained embeddings with dynamic expert selection to enhance robustness and interpretability in clinical AI applications targeting AMR.
- **Enhanced robustness on imbalanced data:** MoEPH achieves stable performance even under highly imbalanced conditions, avoiding overfitting and reliably predicting under-represented bacteria.
- **Interpretability via expert weight visualization:** The gating network output offers interpretable insights into model decisions, supporting clinician trust and biological discovery.
- **Generalization across datasets and novel pairs:** Extensive evaluation shows that MoEPH generalizes well to external datasets and unseen phage–host pairs, maintaining reliability as scientific knowledge evolves.

2 Definition and materials

In pursuit of trustworthy AI, this chapter details the feature design and data preparation that form the foundation for MoEPH, with a particular emphasis on explainability and robustness. Specifically, we integrate knowledge-driven statistical descriptors with context-rich transformer-based protein embeddings to balance interpretability and predictive performance. Fundamental features such as amino acid composition (AAC), atomic composition (AC), and molecular weight (MW) provide human-understandable sequence characteristics that complement the deep sequence representations from pretrained models

¹ CNSA accession number: CNP0006217, available at <https://db.cngb.org/search/project/CNP0006217/>.

(e.g., ProtBERT, ProT5). By unifying these two classes of features within a Mixture-of-Experts (MoE) framework, our model can more effectively capture protein complexity in an adaptive manner. The synergy between interpretable statistical features and advanced LLM-derived embeddings thus serves as the basis for a more explainable and resilient predictive system. In this section, we describe the designed statistical features and the datasets utilized, which together underpin the trustworthy modeling approach of MoEPH.

2.1 Definition of statistical features

We incorporate three interpretable, domain-knowledge-driven statistical descriptors to complement deep embeddings:

(1) **Amino acid composition (AAC)**: Measures the frequency of each amino acid A_i in sequence S :

$$AAC_i = \frac{n_i}{L}, \quad i = 1, \dots, M,$$

where n_i is the count of A_i and L is the sequence length. AAC provides coarse but robust information for classification tasks.

(2) **Atomic composition (AC)**: Represents the proportion of each element E_j (e.g., C, H, N, O, S) in a protein:

$$AC_j = \frac{a_j}{\sum_{k=1}^E a_k}, \quad j = 1, \dots, E,$$

where a_j is the number of atoms of E_j in S . AC captures the protein's fundamental chemical makeup.

(3) **Molecular weight (MW)**: Summarizes compositional information into a physicochemical metric:

$$MW(S) = \sum_{i=1}^M n_i \cdot M(A_i),$$

where $M(A_i)$ denotes the molecular weight of amino acid A_i . MW can discriminate proteins based on mass profiles.

These statistical features enhance model explainability by providing intuitive biochemical insights into protein sequences.

2.2 Transformer-based LLMs protein representations

We leverage ProtBERT and ProT5, two pre-trained protein LLMs, to generate context-rich sequence embeddings.

(1) **ProtBERT**: ProtBERT is a BERT-based protein language model that employs a bidirectional Transformer encoder with a masked language modeling (MLM) objective [20]. During pre-training, a subset $M = \{m_1, \dots, m_K\}$ of positions in the input sequence S is randomly masked, and the model learns to predict the masked residues using context. The MLM loss is defined as:

$$L_{\text{MLM}}(\theta) = - \sum_{m \in M} \log P(s_m | S_{\setminus m}; \theta), \quad (1)$$

where $S_{\setminus m}$ denotes S with the residue at position m replaced by a mask token. ProtBERT stacks multiple Transformer layers to iteratively refine the sequence representation:

$$H^{(l)} = \text{TransformerLayer}(H^{(l-1)}), \quad l = 1, 2, \dots, L, \quad (2)$$

with $H^{(0)}$ being the input token embeddings. By leveraging contextual cues, ProtBERT learns both local motifs and long-range dependencies, yielding robust sequence features for downstream tasks.

(2) **ProT5**: ProT5 is a T5-based encoder-decoder model that frames protein modeling tasks in a sequence-to-sequence format. It can be pre-trained to reconstruct corrupted sequences, predict functional or structural annotations, or even generate novel protein sequences. Given an input sequence S and a target sequence $Y = \{y_1, \dots, y_T\}$, ProT5 models the conditional probability:

$$P(Y | S; \theta) = \prod_{t=1}^T P(y_t | y_{<t}, S; \theta). \quad (3)$$

It learns by minimizing the negative log-likelihood:

$$L_{\text{T5}}(\theta) = - \sum_{t=1}^T \log P(y_t | y_{<t}, S; \theta). \quad (4)$$

The encoder first produces a hidden representation:

$$H_{\text{enc}} = \text{Encoder}(S). \quad (5)$$

The decoder then uses H_{enc} (together with previously generated tokens $y_{<t}$) to predict the next token y_t :

$$H_{\text{dec},t} = \text{Decoder}(H_{\text{enc}}, y_{<t}), \quad t = 1, 2, \dots, T. \quad (6)$$

Through this process, ProT5 captures rich long-range dependencies and excels in generative and multi-task settings. Training on diverse objectives endows ProT5 with a broad latent space of protein sequences, complementing ProtBERT's embeddings.

2.3 Dataset of the immersion experiment

The datasets originate from a BGI research project titled *Research on Artificial Phage Model Construction Based on Deep Generative Adversarial Network Learning* (see text footnote 1), providing a reliable empirical foundation for our prediction tasks. This study is part of an ongoing initiative at BGI-Shenzhen to explore phage-host prediction under real-world data constraints. The equipment is based on the NVIDIA A100 cloud platform. To evaluate the model under multi-source conditions, we utilize two distinct datasets and a third integrated dataset. To evaluate the model under multi-source conditions, we utilize two distinct datasets and a third integrated dataset.

To provide a clearer picture of the dataset composition, we include a heatmap of the host-phage interaction matrix (Figure 1)

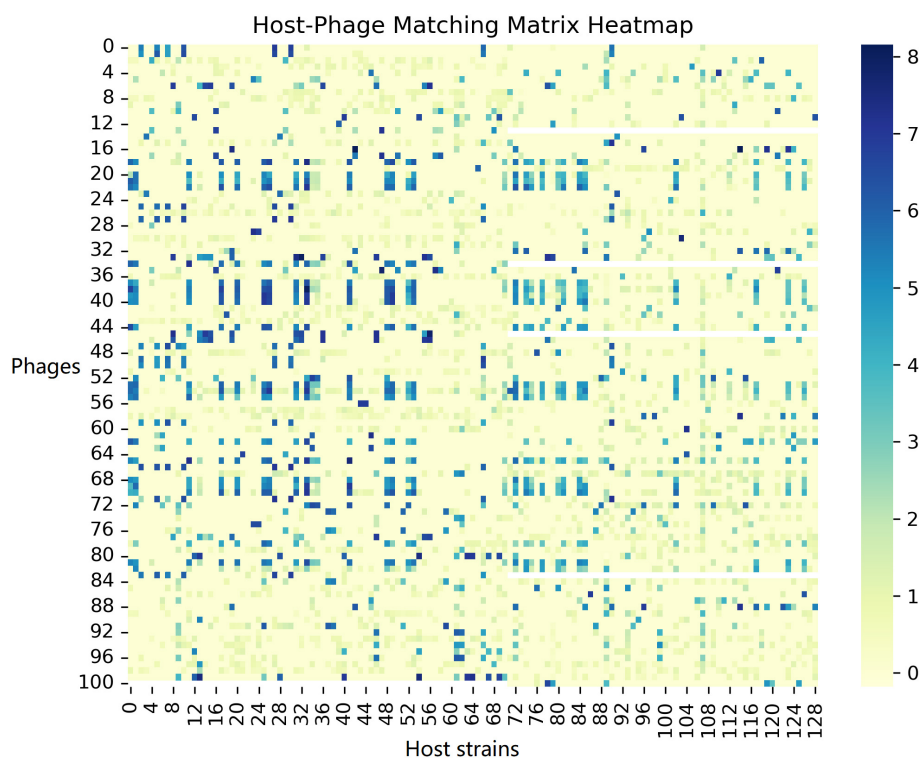


FIGURE 1
Host-phage matching matrix.

in the data description. This figure offers an overview of which phages infect which host strains in Dataset 1, highlighting the data's structure before any modeling.

Dataset 1: This dataset includes 101 host bacterial strains and 129 phages, collected under well-defined conditions, ensuring clear information on species composition and environmental context.

Dataset 2: This dataset comprises 38 host strains and 176 phages. Compared to Dataset 1, its sampling conditions (e.g., environment and strain selection) differ, yielding distinct phage–host interaction patterns.

Dataset 3: Dataset 3 combines Dataset 1 and Dataset 2 to enable performance assessment under more diverse conditions, providing a robust test of the model's adaptability to heterogeneous data sources in real-world settings.

(1) *Immersion experiment and labeling strategy:* The phage–host interactions were measured via immersion experiments, where phages were exposed to host cultures and infection outcomes recorded in a host–phage matrix (Figure 1). We binarized these outcomes by labeling interactions with infection values above 1.5 as 1 (significant infection) and below 1.5 as 0 (no significant infection), ensuring a clear separation between positive and negative samples.

(2) *Sequence information and feature extraction:* For each host and phage, we obtained the protein amino acid sequence and extracted features including traditional descriptors (AAC, AC, MW) and LLM-based embeddings (ProtBERT, ProT5), yielding a rich feature set for learning.

In summary, these curated multi-source datasets and comprehensive feature sets provide a solid foundation to assess the model's generalization and robustness. Their diversity underscores the importance of MoEPH's dynamic fusion mechanism for reliable predictions across heterogeneous conditions.

3 Proposed framework: MoEPH

In this section, we present the architecture of the proposed Mixture-of-Experts model for Phage Host prediction (MoEPH). The MoEPH framework is designed to leverage transformer-based protein representations and an ensemble of expert sub-models to address the complex task of phage–host prediction. Mixture-of-Experts (MoE) architectures have also been used in recent large language models to achieve greater scalability by dividing the model's knowledge among specialized sub-networks; we adopt a similar principle here to effectively handle the heterogeneity of phage–host data. By decomposing the prediction task among multiple expert networks and using an adaptive gating mechanism to fuse their outputs, MoEPH can capture diverse patterns in the data. This design enhances the model's robustness and flexibility, as each expert can specialize in certain features or sub-distributions of the input, while the gating network dynamically selects and combines expert contributions appropriate for each phage query. In what follows, we detail the overall model framework, the structure

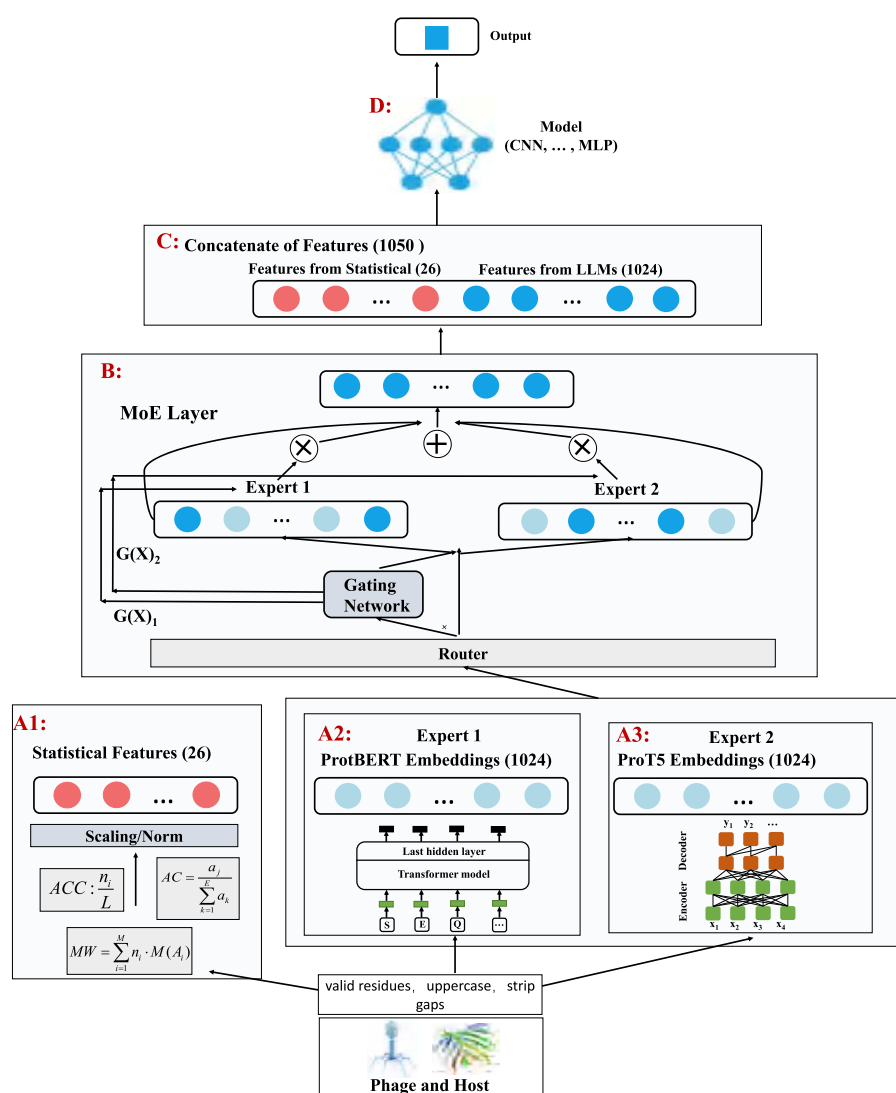


FIGURE 2

Flowchart of MoEPH. This figure illustrates the model's main components, including statistical feature extraction (A1), two Transformer-based LLM feature extraction modules (A2, A3), the MoE layer (B) for adaptive expert weighting, feature concatenation (C), and the final prediction model (D).

of the Mixture-of-Experts layer, and the training and inference procedures.

3.1 MoEPH model framework

The MoEPH model proposed in this study is designed to integrate multi-source features both statistical descriptors and deep sequence embeddings extracted by large language models (LLMs) and to adaptively weight these features through a Mixture-of-Experts (MoE) mechanism. By doing so, the model produces more robust and expressive representations for subsequent classification tasks. Figure 2 provides an overview of the framework.

In A1 (Statistical feature extraction), we derive fundamental statistical descriptors such as amino acid composition (AAC), atomic composition (AC), and molecular weight (MW) from each protein sequence, denoted as $\mathbf{X}_{\text{stats}} \in \mathbb{R}^{N \times 26}$ in the example

illustration. These features reflect basic physicochemical properties of the protein and serve as an initial numeric reference for subsequent integration.

Concurrently, in A2 and A3 (LLM feature extraction), we employ two pre-trained protein language models: ProtBERT and ProT5 to capture deep semantic representations of the sequences, resulting in $\mathbf{X}_{\text{ProtBERT}} \in \mathbb{R}^{N \times 1024}$ and $\mathbf{X}_{\text{ProT5}} \in \mathbb{R}^{N \times 1024}$. These high-dimensional embeddings encode contextual, local structural, and functional information within the sequences, thereby enriching the predictive power for phage-host interaction tasks.

Next, these three kinds of features are fed into B: the MoE layer, which consists of a Router and two expert modules (Expert 1 and Expert 2). A gating network processes the statistical features $\mathbf{X}_{\text{stats}}$ to generate weights α , dictating how $\mathbf{X}_{\text{ProtBERT}}$ and $\mathbf{X}_{\text{ProT5}}$ should be combined. This adaptive weighting ensures that the model can dynamically select the most discriminative or best-matched expert features for each sample under varying data distributions and

Input:

Training sets: $X_{\text{train_ProtBERT}} \in \mathbb{R}^{N_{\text{train}} \times 1024}$,
 $X_{\text{train_ProT5}} \in \mathbb{R}^{N_{\text{train}} \times 1024}$, $X_{\text{train_stats}} \in \mathbb{R}^{N_{\text{train}} \times 26}$;
 Test sets: $X_{\text{test_ProtBERT}} \in \mathbb{R}^{N_{\text{test}} \times 1024}$,
 $X_{\text{test_ProT5}} \in \mathbb{R}^{N_{\text{test}} \times 1024}$, $X_{\text{test_stats}} \in \mathbb{R}^{N_{\text{test}} \times 26}$

Output: $X_{\text{train_combined}}$, $X_{\text{test_combined}}$

Step 1: Initialize MoE layer

MOEGating($\text{input_dim} = 2048$, $\text{stats_dim} = 26$, $\text{hidden_dim} = 64$):

- Input:** $X_{\text{ProtBERT}} \in \mathbb{R}^{N \times 1024}$, $X_{\text{ProT5}} \in \mathbb{R}^{N \times 1024}$,
 $X_{\text{stats}} \in \mathbb{R}^{N \times 26}$
- Compute gating weights from X_{stats} :** First layer (hidden):

$$H = \text{ReLU}(X_{\text{stats}}W_1 + b_1), W_1 \in \mathbb{R}^{26 \times 64}, H \in \mathbb{R}^{N \times 64}$$

Second layer (output):

$$G = HW_2 + b_2, W_2 \in \mathbb{R}^{64 \times 2}, G \in \mathbb{R}^{N \times 2}$$

Apply softmax to get gating weights:

$$\alpha = \text{Softmax}(G) \in \mathbb{R}^{N \times 2}$$

- Fusing experts:** Stack the two experts along a new dimension:

$$X_{\text{experts}} = \text{stack}(X_{\text{ProtBERT}}, X_{\text{ProT5}}, \text{dim} = 1) \in \mathbb{R}^{N \times 2 \times 1024}$$

Reshape or broadcast α to match experts dimension:

$$\alpha_{\text{expanded}} = \alpha.\text{unsqueeze}(2) \in \mathbb{R}^{N \times 2 \times 1}$$

Apply the gating weights to each expert channel:

$$X_{\text{weighted}} = X_{\text{experts}} \odot \alpha_{\text{expanded}} \in \mathbb{R}^{N \times 2 \times 1024}$$

Sum over the expert dimension:

$$X_{\text{fused}} = \sum_{e=1}^2 X_{\text{weighted}}[:, e, :] \in \mathbb{R}^{N \times 1024}$$

Step 2: Compute fused features for training and test sets

$$X_{\text{train_fused}} \leftarrow \text{moe_layer}(X_{\text{train_ProtBERT}}, X_{\text{train_ProT5}}, X_{\text{train_stats}})$$

$$X_{\text{test_fused}} \leftarrow \text{moe_layer}(X_{\text{test_ProtBERT}}, X_{\text{test_ProT5}}, X_{\text{test_stats}})$$

Step 3: Append stats features

$$X_{\text{train_combined}} \leftarrow [X_{\text{train_fused}}, X_{\text{train_stats}}] \in \mathbb{R}^{N_{\text{train}} \times 2100}$$

$$X_{\text{test_combined}} \leftarrow [X_{\text{test_fused}}, X_{\text{test_stats}}] \in \mathbb{R}^{N_{\text{test}} \times 2100}$$

return $X_{\text{train_combined}}$, $X_{\text{test_combined}}$

Algorithm 1. Mixture-of-Experts (MoE) layer feature fusion in MoEPH.

feature patterns. In addition, the resulting weighting coefficients can provide interpretability regarding the relative importance of each LLM for different types of samples.

As depicted in **C (Concatenate of features)**, the fused features obtained from the MoE layer are concatenated with the statistical features, producing $X_{\text{combined}} \in \mathbb{R}^{N \times 1050}$ (illustrative dimensionality). This consolidated vector representation fully integrates both the “protein statistical attributes” and the “LLM-based deep features.”

Finally, **D (Model)** processes X_{combined} using a customizable prediction network (e.g., a CNN, MLP, or RNN) to output the phage-host interaction results. In this study, we employ a CNN classifier to systematically examine how the fused features improve performance (e.g., accuracy, F1-score, AUC) under a fixed network structure.

In summary, MoEPH incorporates statistical features from traditional analyses and advanced embeddings from ProtBERT/ProT5, then uses an adaptive MoE layer to effectively and interpretably combine multi-source information. This integrated solution provides a robust yet flexible approach to phage-host interaction prediction.

3.2 MoE layer of the MoEPH model

In protein-related prediction tasks, relying solely on a single pre-trained model (e.g., ProtBERT or ProT5) often fails to fully capture the diverse sequence patterns and structural information inherent in biological data. To address this limitation, we introduce a Mixture-of-Experts (MoE) mechanism into our model. By combining multiple pre-trained experts and adaptively assigning their importance based on sample-specific statistical attributes, the MoE layer flexibly merges various feature advantages, thereby enhancing predictive performance and overall generalization (Shazeer et al., 2017; Pearce and Zhang, 2021). The entire fusion procedure is summarized in Algorithm 1.

3.3 MoE layer structure and gating mechanism

In our Mixture-of-Experts (MoE) layer, a *gating network* dynamically computes sample-specific weights to fuse the outputs of two expert models (ProtBERT and ProT5). Formally, for each sample i , let $X_{\text{ProtBERT},i} \in \mathbb{R}^d$ and $X_{\text{ProT5},i} \in \mathbb{R}^d$ denote the pre-trained feature vectors from ProtBERT and ProT5 (with $d = 1024$ in our implementation). Each sample also has an associated statistical feature vector $X_{\text{stats},i} \in \mathbb{R}^s$ (with $s = 26$ descriptors such as physicochemical properties and sequence composition). The gating function $g(\cdot)$ is realized as a two-layer feed-forward network that transforms the statistical features into a pair of importance weights $(\alpha_{i,1}, \alpha_{i,2})$ for the two experts. Concretely, the gating network first applies a fully-connected layer to $X_{\text{stats},i}$ (shared across samples) to produce a hidden representation, then a second fully-connected layer produces two logit scores which are

Pretrained feature matrices:
 $X_{\text{ProtBERT}} \in \mathbb{R}^{N \times d}$, $X_{\text{ProT5}} \in \mathbb{R}^{N \times d}$;
Input: Statistical feature matrix:
 $X_{\text{stats}} \in \mathbb{R}^{N \times s}$
Output: $X_{\text{fused}} \in \mathbb{R}^{N \times d}$: fused feature matrix
 $X_{\text{combined}} \in \mathbb{R}^{N \times (d+s)}$;
Step 1: Compute gating weights (sample-specific)
 $H \leftarrow \text{ReLU}(X_{\text{stats}}W_1 + b_1)$ // $H \in \mathbb{R}^{N \times 64}$
 $G \leftarrow HW_2 + b_2$ // $G \in \mathbb{R}^{N \times 2}$
 $\alpha \leftarrow \text{softmax}(G)$ // $\alpha \in \mathbb{R}^{N \times 2}$, each row $\alpha_{i,*}$ sums to 1
Step 2: Fuse expert outputs using gating weights
for $i = 1$ **to** N **do**
 $X_{\text{fused},i} \leftarrow \alpha_{i,1} X_{\text{ProtBERT},i} + \alpha_{i,2} X_{\text{ProT5},i}$
 // $X_{\text{fused},i} \in \mathbb{R}^d$
end
 $X_{\text{fused}} \leftarrow [X_{\text{fused},1}; X_{\text{fused},2}; \dots; X_{\text{fused},N}]$
// $X_{\text{fused}} \in \mathbb{R}^{N \times d}$
Step 3: Concatenate fused features with original stats
 $X_{\text{combined}} \leftarrow [X_{\text{fused}}, X_{\text{stats}}]$ // $X_{\text{combined}} \in \mathbb{R}^{N \times (d+s)}$
return X_{fused} and X_{combined}

Algorithm 2. Pseudocode of MoE gating and expert fusion.

normalized by a softmax activation:

$$H = \text{ReLU}(X_{\text{stats}}W_1 + b_1), \quad H \in \mathbb{R}^{N \times 64}, \quad (7)$$

$$G = HW_2 + b_2, \quad G \in \mathbb{R}^{N \times 2}, \quad (8)$$

$$\alpha = \text{softmax}(G), \quad \alpha \in \mathbb{R}^{N \times 2}, \quad (9)$$

where $W_1 \in \mathbb{R}^{s \times 64}$, $b_1 \in \mathbb{R}^{64}$, $W_2 \in \mathbb{R}^{64 \times 2}$, $b_2 \in \mathbb{R}^2$ are trainable parameters. The ReLU activation in Equation 7 introduces nonlinearity into the gating function, and the softmax in Equation 9 ensures each sample's two gating coefficients ($\alpha_{i,1}, \alpha_{i,2}$) form a probability distribution (i.e., $\alpha_{i,1}, \alpha_{i,2} \geq 0$ and $\alpha_{i,1} + \alpha_{i,2} = 1$ for each i). Importantly, this gating is *dynamic and sample-specific*: for each sample i , the statistical attribute vector $X_{\text{stats},i}$ yields its own gating weights $\alpha_{i,1}$ and $\alpha_{i,2}$. This design allows the model to adaptively decide how much to rely on each expert's features based on the characteristics of that sample (rather than using fixed static fusion weights).

Using the gating weights $\alpha_{i,1}$ and $\alpha_{i,2}$, the MoE layer modulates and fuses the expert outputs for each sample. Let $\alpha_i = [\alpha_{i,1}, \alpha_{i,2}]$ be the weight vector for sample i . We obtain the *fused* feature for sample i by an element-wise weighted sum of the two expert feature vectors:

$$X_{\text{fused},i} = \alpha_{i,1} X_{\text{ProtBERT},i} + \alpha_{i,2} X_{\text{ProT5},i}, \quad X_{\text{fused},i} \in \mathbb{R}^d. \quad (10)$$

In other words, the ProtBERT embedding is scaled by $\alpha_{i,1}$ and the ProT5 embedding by $\alpha_{i,2}$, and then they are added together to produce a single fused representation for sample i . Stacking these results for all N samples yields the fused feature matrix:

$$X_{\text{fused}} = \alpha_1 \odot X_{\text{ProtBERT}} + \alpha_2 \odot X_{\text{ProT5}} \in \mathbb{R}^{N \times d}, \quad (11)$$

where $\alpha_{:,1}, \alpha_{:,2} \in \mathbb{R}^{N \times 1}$ denote the two columns of α (broadcasted across the d -dimensional feature vectors), and $\odot$ denotes element-wise (Hadamard) product. Through Equations 10–11, the gating weights effectively *modulate* the contribution of each expert: if $\alpha_{i,1} \gg \alpha_{i,2}$ for a given sample, the fused representation $X_{\text{fused},i}$ will be dominated by ProtBERT's features, whereas if $\alpha_{i,2}$ is larger, ProT5's features are emphasized. This adaptive fusion flexibly leverages the strengths of both experts, allowing the model to favor the expert that is more informative for each particular sample's attributes.

After obtaining the fused LLM-based features X_{fused} , we integrate them with the original statistical features. Specifically, we concatenate each sample's fused vector with its statistical descriptor vector to form the final combined feature:

$$X_{\text{combined}} = [X_{\text{fused}}, X_{\text{stats}}] \in \mathbb{R}^{N \times (d+s)}. \quad (12)$$

In our implementation $d = 1024$ and $s = 26$, so $X_{\text{combined}} \in \mathbb{R}^{N \times 1050}$. This concatenation preserves the original handcrafted features alongside the fused deep features, ensuring that downstream classifiers receive a comprehensive feature set. The entire gated fusion procedure is summarized in Algorithm 2.

3.4 Model training and prediction

After extracting and preprocessing the features, we employ a convolutional neural network (CNN) as the classification backbone for phage-host interaction prediction, integrating the MoE module for feature fusion throughout the training process. While we fix this CNN architecture for consistency, our main objective is to demonstrate how the proposed MoEPH framework leverages multi-source embeddings to boost predictive performance, the advantage would extend similarly to other model architectures. The overall training and evaluation pipeline is summarized in Algorithm 3.

3.4.1 Network architecture

Let $X \in \mathbb{R}^{N \times d}$ be the combined feature matrix obtained via the MoE fusion layer, where N is the number of samples and d is the dimensionality of the fused features (including both Transformer-based protein embeddings and statistical descriptors). Each sample $x_i \in \mathbb{R}^d$ is reshaped into a $(d, 1)$ array, which is then fed into a series of 1D-CNN layers. Specifically, these layers consist of:

- One-dimensional convolutional layers (e.g., kernel_size = 3, varying channel widths), each followed by a nonlinear activation function (ReLU), batch normalization, and max-pooling;
- Global pooling operation to reduce the final convolution outputs to a fixed-size feature vector;
- Fully connected (FC) layers that project the pooled feature vector into logits for the binary classification task (interaction vs. non-interaction).

Additional regularization elements (e.g., dropout) may be introduced in the FC layers to mitigate overfitting.

Training Sets:
 $X_{\text{train_ProtBERT}}, X_{\text{train_ProT5}},$
Input: $X_{\text{train_stats}}, Y_{\text{train}}$
 Test Sets:
 $X_{\text{test_ProtBERT}}, X_{\text{test_ProT5}},$
 $X_{\text{test_stats}}, Y_{\text{test}}$
Output: Trained CNN model & performance metrics

Step 1: MoE fusion to obtain features
 Use the MoE layer to generate X_{combined} by fusing ProtBERT, ProT5, and the statistical features.

Step 2: Reshape and construct CNN
 1. Reshape:

$$X_{\text{cnn}} \leftarrow \text{Reshape}(X_{\text{combined}}, (N, d, 1)).$$

2. Define the 1D-CNN architecture (e.g., Conv→ReLU→BatchNorm→MaxPool, repeated multiple times, then global pooling, and FC layers).

Step 3: Joint training of MoE & CNN
for $\text{epoch} = 1$ **to** $EPOCHS$ **do**
 $X_{\text{batch_combined}} \leftarrow$
 $\text{MoE}(X_{\text{batch_ProtBERT}}, X_{\text{batch_ProT5}}, X_{\text{batch_stats}})$
 $X_{\text{batch_cnn}} \leftarrow \text{Reshape}(X_{\text{batch_combined}}, (B, d, 1))$
 $\text{pred} \leftarrow \text{cnn_model}(X_{\text{batch_cnn}})$ // Forward pass
 $\text{loss} \leftarrow \text{CrossEntropyLoss}(\text{pred}, y_{\text{batch}})$
 Update MoE & CNN parameters by backpropagation:

$$\nabla \leftarrow \frac{\partial \text{loss}}{\partial (\text{MoE params}, \text{CNN params})}$$

Optimizer.step(∇)

end

Step 4: Evaluation
 1. Obtain $X_{\text{test_combined}}$ via MoE, reshape to $(N_{\text{test}}, d, 1)$.
 2. Predict:

$$\text{pred}_{\text{test}} \leftarrow \text{cnn_model}(X_{\text{test_cnn}})$$

3. Compute performance metrics (Accuracy, F1, AUC-ROC, AUPR, etc.).

return Trained CNN model and performance metrics

Algorithm 3. Model training and evaluation (CNN with MoE fusion).

3.4.2 Loss function

We adopt the cross-entropy loss for binary classification:

$$L_i = -[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)], \quad (13)$$

where $y_i \in \{0, 1\}$ is the true label for sample i , and $\hat{p}_i$ is the predicted probability that sample i is positive (i.e., displays phage-host interaction). Concretely, let $z_{i,0}$ and $z_{i,1}$ be the logits for the

negative and positive classes, respectively, so that

$$\hat{p}_i = \frac{\exp(z_{i,1})}{\exp(z_{i,0}) + \exp(z_{i,1})}. \quad (14)$$

Averaging over the entire training set of size N yields

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]. \quad (15)$$

Minimizing $L(\theta)$ with respect to the model parameters θ steers the CNN toward more accurate probability estimates.

3.4.3 Training procedure

We initialize the CNN parameters θ randomly and adopt the Adam optimizer with a selected learning rate (e.g., 1×10^{-3}). Training proceeds in mini-batches over a fixed number of epochs. For each mini-batch:

1. We obtain the fused features $X_{\text{batch_combined}}$ via the MoE layer.
2. Reshape them to $(B, d, 1)$ for the 1D-CNN, where B is the mini-batch size.
3. Perform a forward pass through the CNN to compute logits and subsequently derive predicted probabilities.
4. Compute the cross-entropy loss using the predicted probabilities and the ground truth labels.
5. Backpropagate to compute gradients $\nabla_{\theta} L(\theta)$ and update all relevant parameters (MoE and CNN) in one unified step.

This joint optimization ensures that the CNN learns discriminative features while the MoE layer continues to refine the expert-selection gating.

3.4.4 Prediction

After convergence, we apply the trained model to the test set. Specifically, the MoE layer fuses the expert embeddings for each test sample, the fused feature vectors are reshaped for the CNN, and the final probabilities $\hat{p}_j$ are obtained via a Softmax layer. We use a threshold of 0.5 to determine the predicted class:

$$\hat{y}_j = \begin{cases} 1, & \text{if } \hat{p}_j \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

Standard classification metrics (accuracy, F1-score, AUC-ROC, AUPR, etc.) are then computed to evaluate predictive performance.

4 Experimental results

4.1 Data description and sampling strategies

This study evaluates the proposed model on three datasets (Dataset1, Dataset2, and a merged Dataset3 which combines the former two). Dataset 1 contains phage-host pairs collected under relatively consistent experimental conditions (with well-defined species compositions and environmental factors), whereas Dataset 2 comes from a more complex ecological background, yielding

greater heterogeneity in phage–host interaction patterns. By merging these two sources into Dataset 3, we impose more stringent demands on the model's adaptability to heterogeneous, multi-source inputs. This strategy was chosen to preserve each dataset's unique characteristics and to assess whether the model generalizes across different sources. Had we merged the datasets from the beginning, any source-specific patterns or performance differences would be hidden. By first testing on each dataset individually, we can demonstrate MoEPH's robust performance under each condition, and then confirm its adaptability on the merged Dataset 3.

To further challenge the model's robustness, each dataset is examined under three class imbalance settings: the original imbalanced distribution (Raw), an Over-sampling variant, and an Under-sampling variant. In Over-sampling, instances from the minority class are replicated to balance the number of positive and negative samples (e.g., if N_{pos} and N_{neg} denote the counts of positive and negative samples with $N_{pos} < N_{neg}$, additional positive instances are randomly duplicated until $N'_{pos} \approx N_{neg}$). In Under-sampling, the opposite approach is applied: majority-class instances are randomly removed until $N'_{neg} \approx N_{pos}$, thereby equalizing class counts. These three sampling methods simulate varying degrees of class imbalance encountered in real-world scenarios, enabling a comprehensive evaluation of the model's robustness across different data distributions.

4.1.1 Sampling methods overview

To validate our model's performance under different class distributions and assess its generalization capability, we applied three sampling strategies to each dataset (as illustrated in Figure 3):

- **Raw (imbalanced data):** Directly using the original dataset while preserving its natural ratio of positive and negative samples, without any additional sampling.
- **Over-sampling:** Replicating instances from the minority class to balance the number of positive and negative samples. For

instance, if N_{pos} and N_{neg} denote the number of positive and negative samples respectively and $N_{pos} < N_{neg}$, then over-sampling randomly duplicates some positive samples until $N'_{pos} \approx N_{neg}$.

- **Under-sampling:** The opposite approach, which randomly removes part of the majority class to match the minority class size. If $N_{pos} < N_{neg}$, we randomly eliminate some negative samples so that $N'_{neg} \approx N_{pos}$.

These three sampling methods simulate varying degrees of class imbalance encountered in real-world scenarios, offering a more comprehensive evaluation of the model's adaptability and generalization performance across different data distributions.

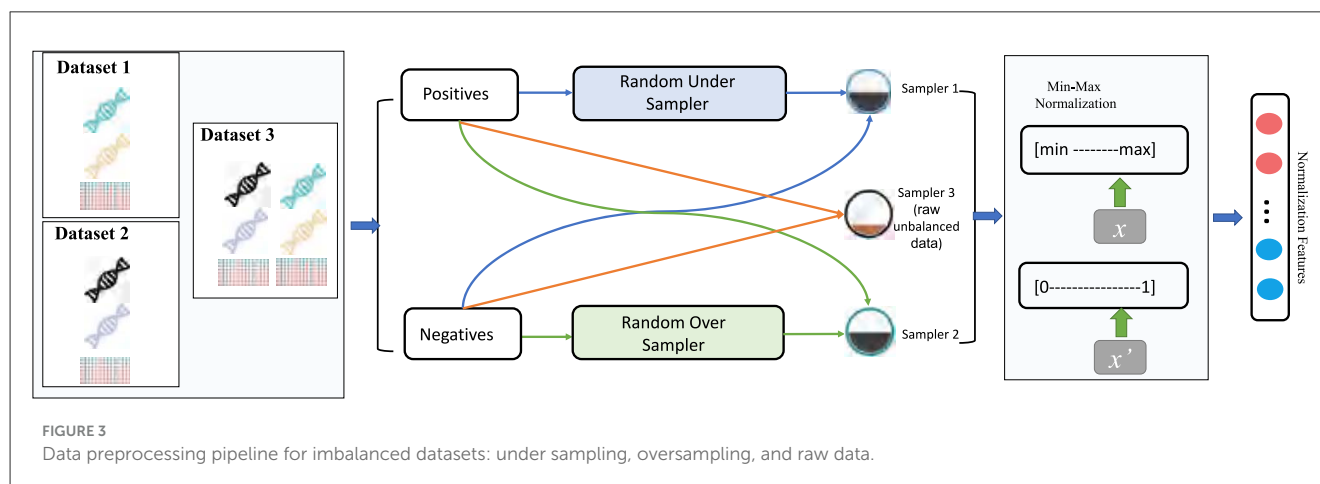
4.1.1.1 Min-max normalization

In addition to resampling, we apply a uniform preprocessing to all features. Min-Max Normalization: To alleviate discrepancies in feature value ranges, each feature is rescaled to $[0, 1]$ via:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (17)$$

where x is an original feature value and x' is its normalized counterpart. This normalization expedites model convergence and enhances stability, especially when combining features of different scales. Finally, to assess the Generality and Applicability of our approach, we design experiments from multiple perspectives:

- **Class distribution impact:** We compare model performance on naturally imbalanced data (Raw) versus balanced data (Over-sampled or Under-sampled) to gauge robustness to uneven class distributions.
- **Multi-source backgrounds:** By evaluating the algorithms on Dataset1, Dataset2, and the combined Dataset 3, we examine generalization under different biological settings and mixed conditions. This tests how well the model adapts to multi-source data variability, highlighting its robustness in a heterogeneous scenario.



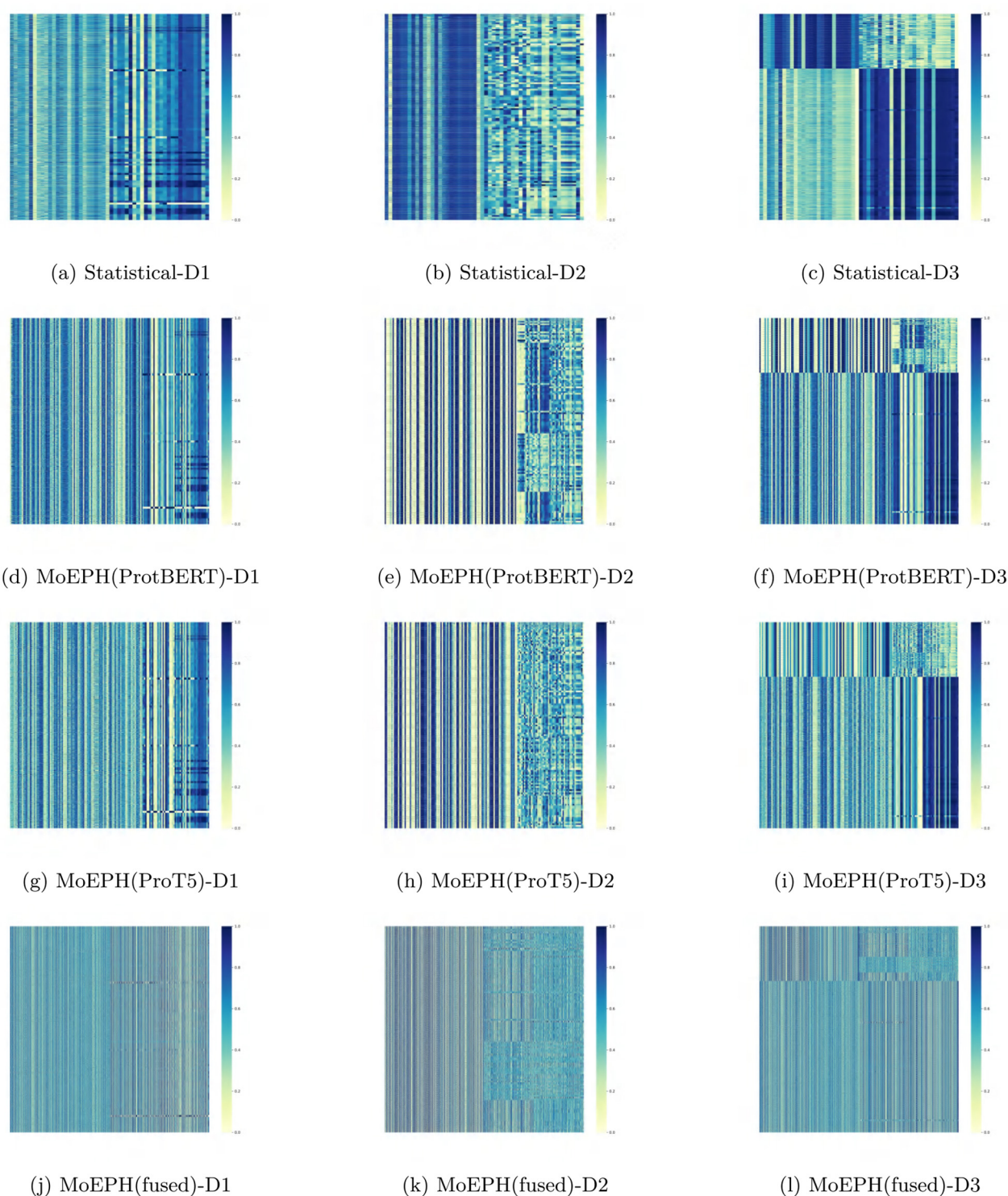


FIGURE 4

Heatmap of features in algorithms on different datasets. (a) Statistical-D1. (b) Statistical-D2. (c) Statistical-D3. (d) MoEPH(ProtBERT)-D1. (e) MoEPH(ProtBERT)-D2. (f) MoEPH(ProtBERT)-D3. (g) MoEPH(ProT5)-D1. (h) MoEPH(ProT5)-D2. (i) MoEPH(ProT5)-D3. (j) MoEPH(fused)-D1. (k) MoEPH(fused)-D2. (l) MoEPH(fused)-D3.

- **Feature visualization:** Although the heatmaps (e.g., Figure 4) primarily reflect the distribution of final fused features rather than changes before sampling, contrasting the feature patterns

generated by different datasets and algorithms still offers useful insights into what aspects the model focuses on and how those relate to potential biological interpretations.

4.2 Performance evaluation metrics

In this study, we adopt several widely recognized metrics to evaluate the predictive capability of our model under class-imbalanced conditions (Fawcett, 2006; Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015; Huang and Ling, 2005). Below, we provide their mathematical definitions and briefly discuss their relevance in the context of phage–host interaction prediction.

$$\text{ACC} = \frac{TP + TN}{TP + FP + TN + FN} \quad (18)$$

The proportion of all predictions that are correct.

$$\text{Prec} = \frac{TP}{TP + FP} \quad (19)$$

The proportion of predicted positive cases that are truly positive.

$$\text{Spec} = \frac{TN}{TN + FP} \quad (20)$$

The proportion of actual negative cases correctly identified (true negative rate).

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

The harmonic mean of precision and recall.

$$\text{AUC} = \int_0^1 \text{TPR}(f) df \quad (22)$$

The area under the ROC curve (TPR vs. FPR), summarizing performance across all thresholds.

$$\text{AUPR} = \int_0^1 \text{Precision}(r) dr \quad (23)$$

The area under the precision–recall curve, reflecting the precision–recall trade-off.

4.3 Feature representation analysis

Figure 4 provides a comparative visualization of the final feature representations produced by different methods across the three datasets. Each vertical column corresponds to a specific feature channel, and the color intensity indicates the magnitude of the feature values (with darker shades representing higher values and lighter shades indicating lower values). These heatmaps are derived from features output by our MoEPH model after the expert fusion layer, just before classification, under various configurations: using only statistical features, using ProtBERT embeddings, using ProT5 embeddings, and using the fused MoE approach (ProtBERT+ProT5). Compared to the traditional statistical baseline, the MoEPH-based models display more distinct block and stripe patterns, indicating that MoEPH captures finer-grained, context-dependent sequence features than conventional methods. Each subfigure corresponds to one method on one dataset (Figures 4a–l), allowing side-by-side comparison of how feature distributions vary by method and data complexity. The key observations are as follows:

4.3.1 Statistical method vs. MoEPH-based features

Compared to the “Statistical Method” (see Figures 4a–c), the MoEPH variants (ProtBERT only, ProT5 only, and the fused approach) typically yield more distinct vertical stripes or block patterns in their heatmaps, with sharper gradients across various feature columns. Because the Statistical Method relies on basic statistical measurements, its derived feature vectors often exhibit relatively homogeneous distribution patterns. By contrast, the MoEPH approaches, having leveraged large pre-trained models, are better able to capture fine-grained, context-dependent sequence representations, leading to more pronounced differences among samples.

4.3.2 Differences between ProtBERT and ProT5

As shown in Figures 4d–i, using ProtBERT versus ProT5 for feature extraction can produce noticeably different heatmap patterns in certain feature columns. In some dimensions, ProtBERT’s response appears more “striped,” whereas ProT5 may display broader regions of lighter or darker shades. This indicates that the two pre-trained models have distinct sensitivities or biases when encoding the same sequence information. Such disparities become even more pronounced for Dataset 2 and Dataset 3, suggesting each model exhibits unique strengths in capturing sequence features under more complex biological conditions.

4.3.3 Performance of the dated fusion [MoEPH (fused)]

Figures 4j–l present the heatmaps of features obtained by gating and fusing the ProtBERT and ProT5 experts. These images reveal that while certain ProtBERT-like or ProT5-like textures remain, some local regions appear smoothed out or amplified. This indicates that the gating mechanism is not merely an averaging step but rather a selective weighting process driven by statistical features and sequence context, yielding a more diversified feature distribution in complex data scenarios.

4.3.4 Changes across datasets

A vertical comparison from Dataset 1 through Dataset 3 shows that as the dataset size and heterogeneity increase, the color patterns—blocks and stripes—also become more pronounced. For instance, in Dataset 3, which includes more varied samples, the color intensity for a given feature column may fluctuate more widely across different instances, suggesting that the large pre-trained models have learned broader sequence distinctions. Conversely, if a method is relatively insensitive to environmental variation, its heatmaps may exhibit similar patterns across datasets, implying limited adaptability in its learned feature space.

Overall, these visual observations suggest that MoEPH-based methods manifest more distinct block structures in their features and are better equipped than traditional statistical approaches to capture deeper sequence-level variations—an advantage that can underpin improved classification results in subsequent experiments.

TABLE 1 Performance comparison across three datasets (D1, D2, D3) under three sampling strategies (Raw, Over, Under) with four algorithms [Statistical, MoEPH (ProtBERT), MoEPH (ProtT5), MoEPH (fused)].

Dataset + sampling	Metric	Statistical (baseline)	MoEPH (ProtBERT)	MoEPH (ProtT5)	MoEPH (fused)
D1 - Raw	ACC	0.966	0.968	<u>0.975</u>	0.969
	F1	0.81	0.81	<u>0.852</u>	0.827
	AUPR	0.837	<u>0.89</u>	0.878	0.887
	AUC	0.969	0.975	<u>0.977</u>	0.972
	Sens	0.8	0.743	<u>0.804</u>	0.801
	Spec	0.982	0.99	<u>0.992</u>	0.986
	Prec	0.821	0.891	<u>0.906</u>	0.854
D1 - Over	ACC	0.98	0.98	<u>0.984</u>	0.978
	F1	0.98	0.98	<u>0.984</u>	0.978
	AUPR	0.994	0.987	<u>0.997</u>	0.993
	AUC	0.996	0.994	<u>0.998</u>	0.996
	Sens	<u>0.999</u>	0.998	<u>0.999</u>	0.992
	Spec	0.961	0.962	<u>0.969</u>	0.965
	Prec	0.963	0.963	<u>0.97</u>	0.965
D1 - Under	ACC	0.917	<u>0.964</u>	0.901	0.891
	F1	<u>0.919</u>	0.791	0.905	0.899
	AUPR	<u>0.948</u>	0.828	0.939	0.926
	AUC	0.956	<u>0.966</u>	0.952	0.946
	Sens	0.937	0.761	0.939	<u>0.957</u>
	Spec	0.896	<u>0.984</u>	0.863	0.825
	Prec	<u>0.901</u>	0.824	0.874	0.847
D2 - Raw	ACC	<u>0.988</u>	<u>0.988</u>	0.985	0.985
	F1	<u>0.768</u>	0.749	0.725	0.674
	AUPR	<u>0.707</u>	0.686	0.654	0.684
	AUC	0.875	0.872	<u>0.903</u>	0.864
	Sens	<u>0.716</u>	0.642	0.679	0.541
	Spec	0.996	<u>0.998</u>	0.994	<u>0.998</u>
	Prec	0.83	<u>0.897</u>	0.779	0.894
D2 - Over	ACC	<u>0.997</u>	0.996	0.996	0.996
	F1	0.997	0.996	<u>0.999</u>	0.996
	AUPR	0.999	0.999	<u>1</u>	0.998
	AUC	<u>0.999</u>	<u>0.999</u>	<u>0.999</u>	<u>0.999</u>
	Sens	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>
	Spec	<u>0.994</u>	0.991	0.993	0.992
	Prec	<u>0.994</u>	0.991	0.993	0.992
D2 - Under	ACC	0.53	0.84	<u>0.845</u>	<u>0.845</u>
	F1	0.44	0.826	<u>0.832</u>	0.829
	AUPR	0.577	0.732	0.814	<u>0.859</u>
	AUC	0.526	0.824	0.85	<u>0.899</u>
	Sens	0.393	0.809	<u>0.819</u>	0.798
	Spec	0.65	0.868	0.868	<u>0.887</u>
	Prec	0.5	0.844	0.846	<u>0.862</u>

(Continued)

TABLE 1 (Continued)

Dataset + sampling	Metric	Statistical (baseline)	MoEPH (ProtBERT)	MoEPH (ProT5)	MoEPH (fused)
D3 - Raw	ACC	0.95	0.954	0.947	<u>0.96</u>
	F1	0.663	0.648	0.628	<u>0.7</u>
	AUPR	0.575	0.62	0.581	<u>0.724</u>
	AUC	0.895	<u>0.904</u>	0.902	0.9
	Sens	<u>0.657</u>	0.566	0.6	0.623
	Spec	0.974	0.985	0.975	<u>0.988</u>
	Prec	0.67	0.757	0.658	<u>0.8</u>
D3 - Over	ACC	0.969	0.961	<u>0.977</u>	0.966
	F1	0.97	0.962	<u>0.977</u>	0.967
	AUPR	0.985	0.976	<u>0.988</u>	0.986
	AUC	0.991	0.986	<u>0.993</u>	0.991
	Sens	0.993	0.983	<u>0.996</u>	0.992
	Spec	0.946	0.94	<u>0.958</u>	0.941
	Prec	0.948	0.9415	<u>0.96</u>	0.943
D3 - Under	ACC	0.790	<u>0.793</u>	0.782	0.74
	F1	<u>0.796</u>	0.793	0.789	0.745
	AUPR	0.853	0.831	<u>0.855</u>	0.85
	AUC	<u>0.857</u>	0.851	0.85	0.833
	Sens	<u>0.818</u>	0.798	<u>0.818</u>	0.759
	Spec	0.763	<u>0.788</u>	0.745	0.722
	Prec	0.775	<u>0.789</u>	0.762	0.73

The bold-underlined entries in each row indicate the highest value(s).

4.4 Performance comparison of and analysis

We conducted an in-depth evaluation involving 36 different experimental settings, spanning three datasets, four algorithms, and three data sampling strategies. As shown in [Table 1](#), the MoEPH series—whether incorporating ProtBERT, ProT5, or both—achieved high performance in most scenarios, covering 239 out of 252 metrics (i.e., 94.9%). Its maximum accuracy reached 99.6%, significantly surpassing the current state-of-the-art methods (see [Table 2](#)). Moreover, in more complex datasets, the accuracy (ACC) was improved by as much as 31%, rising from 0.535 to 0.845 ([Figure 5](#)). In some over-sampled scenarios, certain cross-validation folds with very few positive instances yielded a Sensitivity of 1.0. We note that this perfect recall is due to the extremely low number of positives in those folds, reflecting class imbalance.

[Figure 6](#) illustrates the comparison between our proposed MoEPH variants (ProtBERT, ProT5, and fused) and the Statistical approach, as well as PredPHI ([Li et al., 2020](#)), under the Dataset 2 (Under-sampling) scenario. The metrics presented include Accuracy (ACC), Sensitivity (Sens), and Specificity (Spec). We observe that MoEPH (ProtBERT) and MoEPH (ProT5) attain ACC values of 0.84 and 0.845, respectively, outperforming both the Statistical method (0.53) and PredPHI (0.78) by a notable margin. They also exhibit superior Sensitivity, indicating that even under

TABLE 2 Comparison of best accuracy among state-of-the-art methods in phage–host interaction prediction.

Algorithm	Best accuracy (%)
MoEPH (Ours)	99.6%
PredPHI (Li et al., 2020)	81%
Host Phinder (Villarreal et al., 2016)	81%
VirHost Matcher (Ahlgren et al., 2017)	64%
WIsH (Galiez et al., 2017)	63%
LMFH VH (Liu et al., 2018)	63.17%
ILMF VH (Liu et al., 2019)	63.66%
Leite et al. (2018)	95.7%

severe under-sampling, the model can still capture a larger portion of positive samples, thus reducing the rate of missed detections. Furthermore, to comprehensively evaluate performance within the same CNN classification architecture, we expand our discussion to include additional metrics, analyzing the results from three key perspectives: varying protein representation algorithms, different sampling strategies, and multiple datasets.

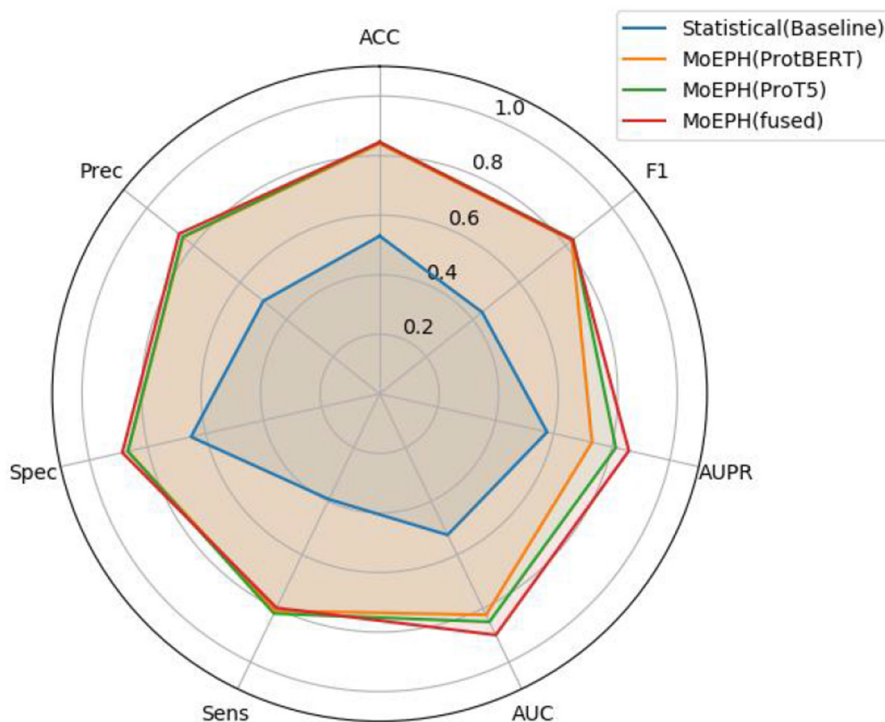


FIGURE 5
Radar chart of metrics in dataset 2 (under sample).

4.4.1 Comparison across different algorithms

4.4.1.1 Single-expert vs. fusion

- MoEPH (ProtBERT) vs. MoEPH (ProT5).** Each model shows its strengths under different conditions. For instance, in **D1-Raw** and **D1-Over**, MoEPH(ProT5) outperforms MoEPH (ProtBERT) in multiple metrics (ACC, F1, Spec, Prec). Conversely, under **D1-Under**, MoEPH (ProtBERT) achieves higher ACC = 0.964, AUC = 0.966, and Spec = 0.984. This discrepancy indicates that the two pretrained models diverge in their focus on protein sequences, likely attributable to differences in training corpora and model architectures.
- MoEPH (fused).** In many scenarios (e.g., D3-Raw), the gating-fusion approach can integrate the merits of both experts and emerge as the best or near-best across multiple metrics (ACC = 0.96, F1 = 0.70, AUPR = 0.724, Prec = 0.80). Nevertheless, there are instances (e.g., D1-Raw) where it slightly lags behind a single expert in some metrics, yet still maintains robust overall performance and often remains on par with or superior to baseline methods. Consequently, gating fusion tends to excel in more complex or highly imbalanced data distributions, whereas a single expert may suffice in simpler scenarios or under near-optimal sampling conditions (e.g., over-sampling).

4.4.1.2 Surprising strength of the statistical baseline

Although the learned embedding features generally produce superior results, the traditional statistical feature baseline exhibited some surprisingly competitive performances in specific cases.

In many comparisons the statistical approach does not achieve the highest scores—highlighting its limited capacity to capture deep semantic cues—yet under certain dataset and sampling configurations it ties or even outperforms the more complex models on some metrics. For example, in D2-Raw, the statistical method attains leading or tied performance with ACC = 0.988 (tied with ProtBERT), F1 = 0.768, AUPR = 0.707, and Sens = 0.716. Likewise, in D1-Under, it achieves the highest F1 = 0.919, AUPR = 0.948, Sens = 0.937, and Prec = 0.901, while MoEPH (ProtBERT) leads in ACC, AUC, and Spec. This “split across metrics” illustrates that, though the statistical method lacks the contextual awareness derived from large-scale pre-training, its more streamlined features can adapt well to particular data distributions or under-sampling schemes, thereby yielding notably effective positive-class recognition in certain cases.

4.4.2 Comparison across sampling methods

4.4.2.1 Over-sampling

In the D1-Over, D2-Over, and D3-Over scenarios, most algorithms achieve extremely high Accuracy, AUC, and even Sensitivity = 1, with differences often only discernible at the third decimal place. Over-sampling balances the number of positive and negative classes by substantially amplifying the positive samples, thus making learning more straightforward for most methods and providing enough data to correct any prior bias toward the negative class.

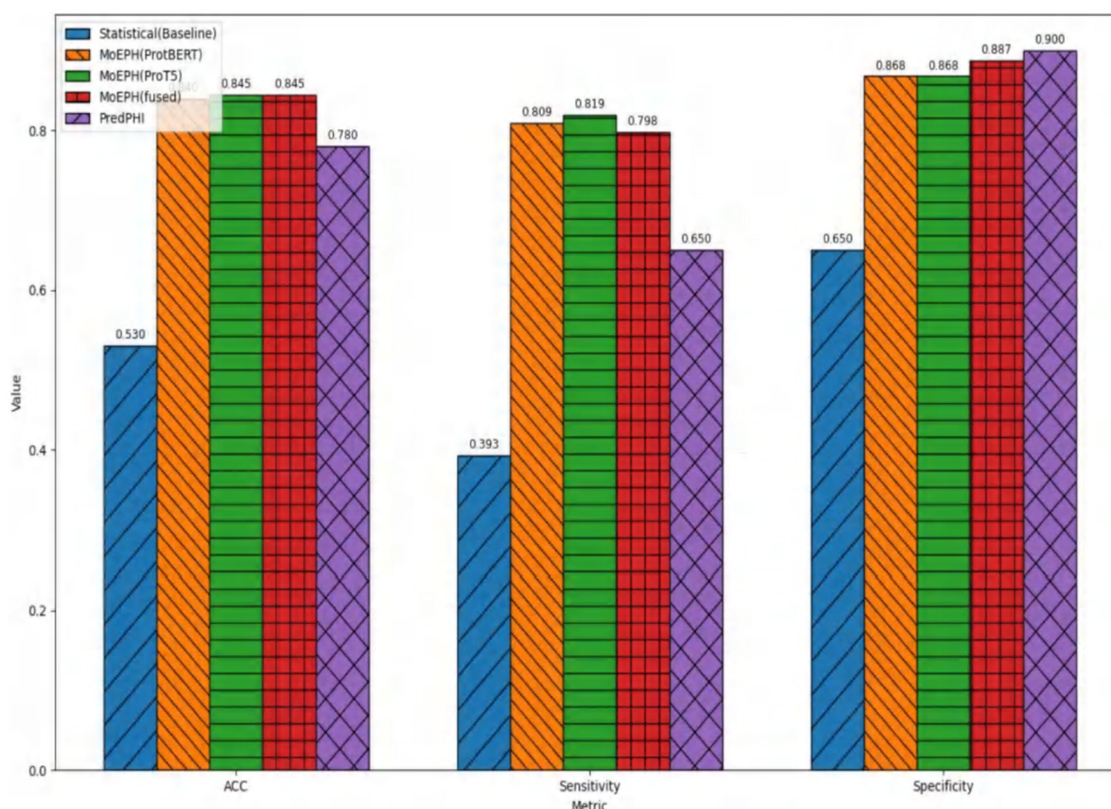


FIGURE 6
Comparison on metrics in Dataset 2 (under sample) with PredPHI (Li et al., 2020).

4.4.2.2 Under-sampling

Under-sampling also balances class counts but does so by heavily removing majority-class samples, which reduces the total amount of available information. For example, in D2-Under, the Statistical method's Accuracy of 0.53 and F1 of 0.44 are conspicuously lower than the MoEPH variants (all exceeding 0.82 in F1). This indicates that large-scale pre-trained representations can maintain discriminative power even under extreme data reduction. Meanwhile, because D1-Under is intrinsically easier to separate, the Statistical approach outperforms single-model variants in certain metrics (F1, AUPR, Sens, Prec), yet still exhibits a noticeable gap in Accuracy. While the statistical model achieves competitive scores on some datasets, its lack of contextual embedding and limited generalization restricts its utility in more diverse prediction settings (Figure 7).

And each cell in Figure 8 shows the percentage of instances (on the test set) belonging to the actual class (rows: Positive or Negative) that were predicted as Positive or Negative (columns). The baseline (a) misses a majority of actual positives (only 40% recall) and produces many false positives (35%), whereas MoEPH (fused) (b) correctly identifies the vast majority of positives (80% recall) while keeping false positives low (12%). This demonstrates MoEPH's significantly improved balance between sensitivity and specificity under extreme class imbalance.

4.4.2.3 Raw

Retaining the natural distribution of the data (Raw) can cause fluctuations in some metrics. Nonetheless, across D1, D2, and D3, the MoEPH series generally demonstrates sufficiently strong performance, suggesting that, under real-world distributions, these methods' differences and applicability become more evident.

4.4.3 Comparison across different datasets

4.4.3.1 D1 vs. D2

D1 appears easier to separate; when over-sampling is applied, nearly all metrics exceed 0.98, reflecting high performance under any sampling approach. In contrast, D2 can reach near-perfect scores under over-sampling (Sensitivity = 1), yet experiences a drastic decline in some algorithms under under-sampling (e.g., Statistical with F1 = 0.44 vs. MoEPH (fused) with F1 = 0.829). This stark contrast indicates that for D2's more complex distribution, substantially removing majority-class samples significantly impairs methods lacking higher-level sequence semantics and contextual awareness. As shown in Figure 8, the Statistical (Baseline) approach fails to converge, whereas MoEPH-based algorithms rapidly reduce loss within the initial epochs. This highlights how pre-trained models and gating fusion can more effectively capture deep protein-sequence features, thereby converging to superior solutions with fewer training iterations.

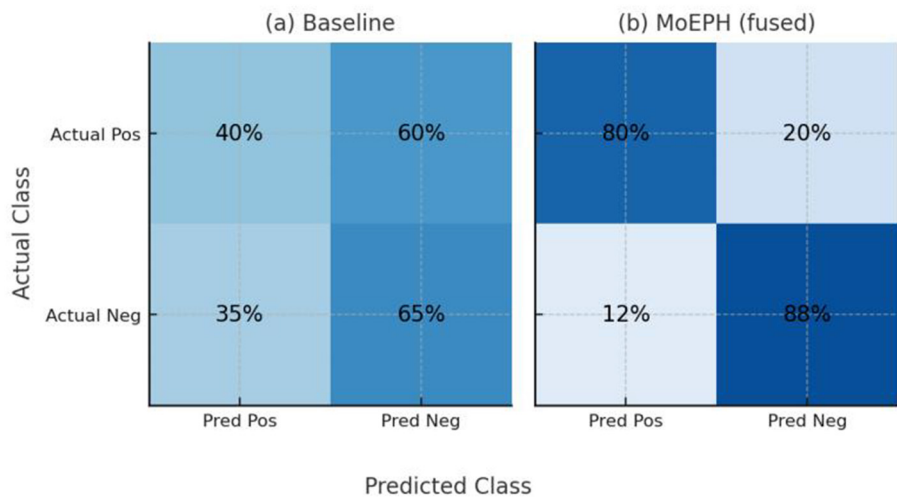


FIGURE 7
Confusion matrices on Dataset2 (Under-sampled) for (a) the Statistical baseline and (b) MoEPH (fused).

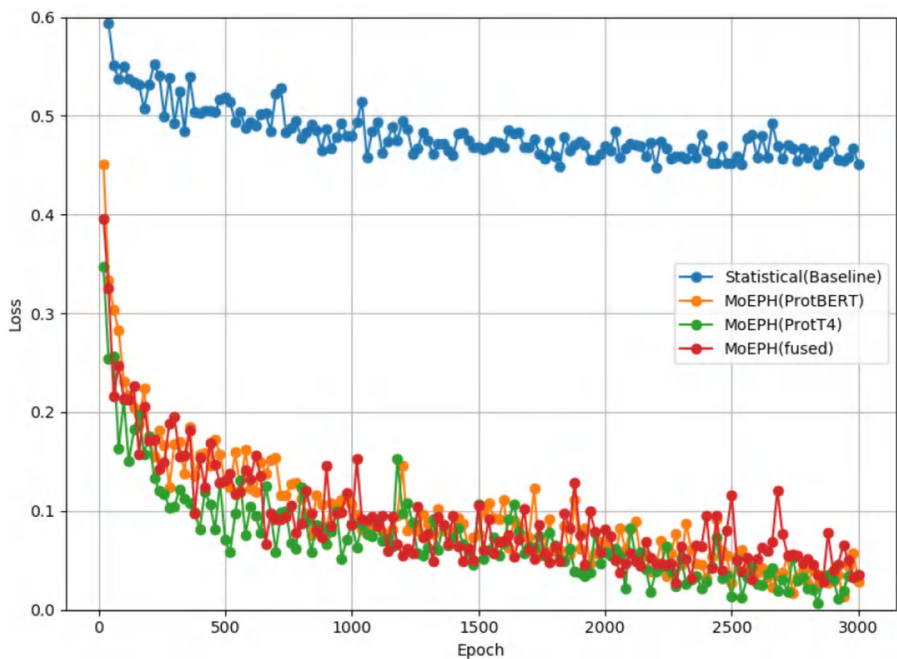


FIGURE 8
Loss curves for four algorithms.

4.4.3.2 D3

D3 integrates or extends the complexities of the previous distributions, yielding results that are neither “near perfect” (as in D2-Over or D1-Over) nor drastically diminished (as in D2-Under). In D3-Raw, MoEPH (fused) achieves F1 = 0.70, AUPR = 0.724, and Prec = 0.80, clearly exceeding Statistical (F1 = 0.663, AUPR = 0.575,

Prec = 0.67), while ProtBERT or ProT5 also excel in selected metrics. This suggests that under broader distributions, large-scale models’ fine-grained sequence representations can further uncover subtle differences. Likewise, D3-Over yields nearly optimal outcomes, minimizing algorithmic discrepancies; in D3-Under, metrics as a whole decline, yet Statistical remains reasonably

competitive (ACC = 0.79, F1 = 0.796, AUPR = 0.853). However, certain MoEPH methods still exhibit minor advantages across other metrics, though the gap is less extreme than in D2-Under. Overall, different datasets impose substantial impact on each algorithm's performance, yet MoEPH consistently outperforms baseline approaches in most metrics.

4.5 Clinical innovation

MoEPH holds promising potential to innovate patient care in the context of antibiotic-resistant infections. By rapidly and accurately predicting phage–host interactions, MoEPH could assist clinicians in selecting effective phage therapies tailored to a patient's drug-resistant bacterial infection, exemplifying precision medicine in infectious disease treatment. This approach could be integrated into the clinical workflow as a decision-support tool, where its robust and interpretable predictions provide physicians with high-confidence recommendations for alternative treatments when antibiotics fail. The model's emphasis on interpretability and reliability builds the trust necessary for clinical adoption, ensuring that healthcare providers can understand and rely on its suggestions. Ultimately, a trustworthy AI system like MoEPH could streamline the management of AMR cases—improving treatment outcomes by offering timely, personalized therapeutic options and potentially integrating into hospital infection control and antibiotic stewardship programs.

5 Conclusion

In this work, we presented MoEPH—a mixture-of-experts model that combines traditional statistical descriptors with deep protein embeddings (ProtBERT and ProT5) to tackle the phage–host prediction problem. Experiments on three benchmark datasets with varied sampling regimes demonstrated that MoEPH consistently outperforms both conventional statistical classifiers and single-model LLM baselines. Notably, MoEPH achieved up to 99.6% accuracy on balanced datasets, and improved accuracy by as much as 31 percentage points on highly imbalanced datasets. The model's adaptive fusion of domain-specific features with pre-trained embeddings ensures robust generalization, while its gating mechanism provides transparency by indicating each expert's contribution to a given prediction. Looking ahead, we plan to further enhance MoEPH along several directions. First, we will incorporate structural protein features (e.g., 3D conformational information) to complement the sequence-based embeddings. Second, we aim to explore alternative neural network architectures as backbones for the expert models, which may uncover additional performance gains. These enhancements are expected to broaden MoEPH's applicability to diverse biomedical prediction tasks, while ensuring the model remains a reliable and transparent AI tool for real-world phage–host identification challenges.

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 authors.

Author contributions

QC: Writing – original draft, Methodology, Data curation, Investigation, Project administration, Formal analysis. ZZ: Data curation, Investigation, Writing – review & editing. ML: Data curation, Writing – review & editing. WS: Investigation, Writing – review & editing. MX: Validation, Visualization, Funding acquisition, Writing – original draft. MF: Writing – original draft, Methodology, Visualization.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was supported by the Beijing Genomics Institute under grant numbers 62176164 and 62203134. National Nature Science Foundation of China Grant 32100130, the datasets is provided by BGI-Shenzhen (CNSA:CNP0006217) and the Scientific Foundation for Youth Scholars of Shenzhen University (868-000001033385).

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.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

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.

References

- Ahlgren, N. A., Ren, J., Lu, Y. Y., Fuhrman, J. A., and Sun, F. (2017). Alignment-free oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic Acids Res.* 45, 39–53. doi: 10.1093/nar/gkw1002
- Aljohani, M., Hou, J., Kommu, S., and Wang, X. (2025). A comprehensive survey on the trustworthiness of large language models in healthcare. *arXiv preprint arXiv:2502.15871*.
- Chen, D., Hartout, P., Pellizzoni, P., Oliver, C., and Borgwardt, K. (2024). Endowing protein language models with structural knowledge. *arXiv preprint arXiv:2401.14819*.
- Coutinho, F. H., Zaragoza-Solas, A., López-Pérez, M., Barylski, J., Zielezinski, A., Dutilh, B. E., et al. (2021). RaFAH: host prediction for viruses of bacteria and archaea based on protein content. *Patterns* 2:100274. doi: 10.1016/j.patter.2021.100274
- Davis, J., and Goadrich, M. (2006). “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd International Conference on Machine Learning*, 233–240. doi: 10.1145/1143844.1143874
- Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., et al. (2021). Prottrans: toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 7112–7127. doi: 10.1109/TPAMI.2021.3095381
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognit. Lett.* 27, 861–874. doi: 10.1016/j.patrec.2005.10.010
- Galiez, C., Siebert, M., Enault, F., Vincent, J., and Söding, J. (2017). Wish: who is the host? Predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics* 33, 3113–3114. doi: 10.1093/bioinformatics/btx383
- Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., et al. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare* 3, 1–23. doi: 10.1145/3458754
- Huang, J., and Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* 17, 299–310. doi: 10.1109/TKDE.2005.50
- Leite, D. M. C., Brochet, X., Resch, G., Que, Y.-A., Neves, A., and Pena-Reyes, C. (2018). Computational prediction of inter-species relationships through omics data analysis and machine learning. *BMC Bioinform.* 19, 151–159. doi: 10.1186/s12859-018-2388-7
- Li, M., Wang, Y., Li, F., Zhao, Y., Liu, M., Zhang, S., et al. (2020). A deep learning-based method for identification of bacteriophage-host interaction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 18, 1801–1810. doi: 10.1109/TCBB.2020.3017386
- Liu, D., Hu, X., He, T., and Jiang, X. (2018). “Virus-host association prediction by using kernelized logistic matrix factorization on heterogeneous networks,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (IEEE), 108–113. doi: 10.1109/BIBM.2018.8621214
- Liu, D., Ma, Y., Jiang, X., and He, T. (2019). Predicting virus-host association by kernelized logistic matrix factorization and similarity network fusion. *BMC Bioinform.* 20, 1–10. doi: 10.1186/s12859-019-3082-0
- Murray, C. J., Ikuta, K. S., Sharara, F., Swetschinski, L., Aguilar, G. R., Gray, A., et al. (2022). Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *Lancet* 399, 629–655. doi: 10.1016/S0140-6736(21)02724-0
- Pan, J., You, Z., You, W., Zhao, T., Feng, C., Zhang, X., et al. (2024). A microbial knowledge graph-based deep learning model for predicting candidate microbes for target hosts. *Brief. Bioinform.* 25:bbae119. doi: 10.1093/bib/bbae119
- Pearce, R., and Zhang, Y. (2021). Deep learning techniques have significantly impacted protein structure prediction and protein design. *Curr. Opin. Struct. Biol.* 68, 194–207. doi: 10.1016/j.sbi.2021.01.007
- Saito, T., and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 10:e0118432. doi: 10.1371/journal.pone.0118432
- Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., et al. (2017). Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.
- Villarroel, J., Kleinheinz, K. A., Jurtz, V. I., Zschach, H., Lund, O., Nielsen, M., et al. (2016). Hostphinder: a phage host prediction tool. *Viruses* 8:116. doi: 10.3390/v8050116
- Walsh, T. R., Gales, A. C., Laxminarayan, R., and Dodd, P. C. (2023). Antimicrobial resistance: addressing a global threat to humanity. *PLoS Med.* 20:e1004264. doi: 10.1371/journal.pmed.1004264
- Wang, W., Ren, J., Tang, K., Dart, E., Ignacio-Espinoza, J. C., Fuhrman, J. A., et al. (2020). A network-based integrated framework for predicting virus-prokaryote interactions. *NAR Genom. Bioinform.* 2:lqaa044. doi: 10.1093/nargab/lqaa044