OPEN ACCESS

ARTICLE

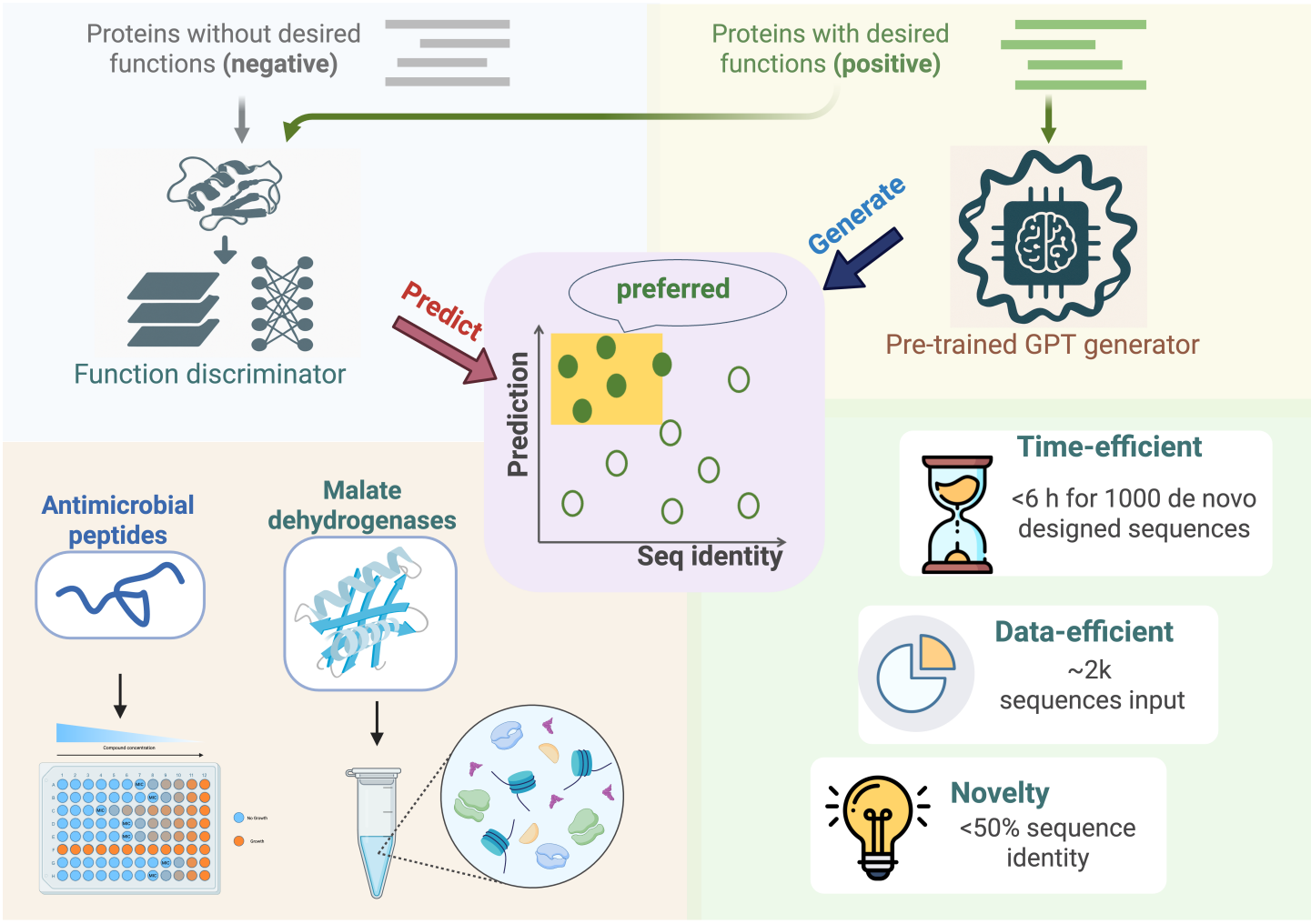
Accelerating functional protein discovery with GPT models: Antimicrobials and enzymes

Zishuo Zeng,[1,*] Rufang Xu,[1,2] Jin Guo,[3] Jiao Jin,[3] Haibing He,[1] and Xiaozhou Luo[1,*]

*Correspondence: zs.zeng@siat.ac.cn (Z.Z.); xz.luo@siat.ac.cn (X.L.)

Received: January 11, 2025; Accepted: April 22, 2025; Published Online: April 25, 2025; https://doi.org/10.59717/j.xinn-life.2025.100133

GRAPHICAL ABSTRACT



PUBLIC SUMMARY

- Finetuning large language model-based protein generator requires only hundreds of protein sequences.
- Diversity of finetuning set affects diversity of generated candidates.
- Prompting is not necessary to guide the generation.
- Coupling a discriminator with protein generator enables efficient design of functional and novel proteins.
- We designed experimentally valid antimicrobial peptides and malate dehydrogenases using this framework.

ARTICLE

OPEN ACCESS

www.the-innovation.org/life

Accelerating functional protein discovery with GPT models: Antimicrobials and enzymes

Zishuo Zeng,[1,*] Rufang Xu,[1,2] Jin Guo,[3] Jiao Jin,[3] Haibing He,[1] and Xiaozhou Luo[1,*]

[1]Key Laboratory of Quantitative Synthetic Biology, Center for Synthetic Biochemistry, Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

[2]College of Chemistry and Pharmacy, Northwest A&F University, Yangling 712100, China

[3]Synceres Biosciences Co. Ltd. 1100 Huanli Rd, Guangming District, Shenzhen 518083, China

*Correspondence: zs.zeng@siat.ac.cn (Z.Z.); xz.luo@siat.ac.cn (X.L.)

Received: January 11, 2025; Accepted: April 22, 2025; Published Online: April 25, 2025; https://doi.org/10.59717/j.xinn-life.2025.100133

Citation: Zeng Z., Xu R., Guo J., et al. (2025). Accelerating functional protein discovery with GPT models: Antimicrobials and enzymes. *The Innovation Life* **3**:100133.

Generative pre-trained transformers (GPT) models provide powerful tools for de novo protein design (DNPD). GPT-based DNPD involves three procedures: (a) finetuning the model with proteins of interest; (b) generating sequence candidates with the finetuned model; and (c) prioritizing the sequence candidates. Existing prioritization strategies heavily rely on sequence identity, undermining the diversity. Here, we coupled a protein GPT model with a custom discriminator, which enabled selecting candidates of low identity to natural sequences while highly likely with desired functions. We applied this framework to creat novel antimicrobial peptides (AMPs) and malate dehydrogenases (MDHs). Experimental verification pinpointed four broad-spectrum AMPs from 24 candidates. Comprehensive computational analyses on the prioritized MDHs candidates provided compelling evidence for the anticipated function. During experimental validation, 4/10 and 3/10 natural MDHs and generated-prioritized novel candidates, respectively, were expressed and soluble. All the soluble candidates (3/3) are functional in vitro. In a broader scope, our generator-discriminator framework is seemingly akin to generative adversarial network (GAN)—but they are fundamentally different. Our results suggest that our framework is more data- and time-efficient than GAN-based method in DNPD and may therefore considerably expedite the DNPD process.

INTRODUCTION

De novo protein design (DNPD), i.e., generating artificial proteins with desired functions, has been proven to be a powerful tool in overcoming bottlenecks across various fields, including drug discovery,[1-3] biosensor engineering,[4-6] and synthetic biology.[7-9] DNPD is being accelerated by recent advancements in generative large language models (LLMs, including GPT),[10] for examples, ProtGPT2[11] and ProGen.[12] Similar to the LLMs for natural language processing (NLP),[13] these protein generative models were trained on large amount of "text corpus" (proteomes) and can generate "sentences" (protein sequences) that captures the "language semantics" (physicochemical and thermodynamic properties similar to natural proteins). Besides, just like LLMs for NLP can be finetuned to adapt to a specific task (such as automatic dialogues,[14] summarizing text,[15] and poetry writing[16]), protein generative models can be finetuned[17] to generate artificial proteins bearing a specific function of interest. Therefore, protein LLM holds great promise in DNPD.

However, evaluating the quality of the generated sequences remains a challenge, as protein sequences are not human-understandable like natural languages. In the work of ProGen,[12] evaluating the generated sequences was mainly done by per-token log-likelihood scoring, which computes the per-residue conditional probabilities along the sequence, conceptually akin to perplexity, i.e., the model's confidence in the generated sequence. However, limitations of perplexity in the NLP field have been noticed, such as sensitivity to text length and repeated content,[18] inability to reflect semantic coherence,[19] and disagreements with human judgements.[20] Another common approach for sequence evaluation is through its identity to know natural proteins of interest, which was used in a generative adversarial network (GAN)-based framework, ProteinGAN.[21] In this approach, the generated sequences are prioritized based on their similarity to the known homologs. Both per-token log-likelihood and sequence identity may overlook the promising (likely to possess the desired protein function) but novel (very different from known natural proteins) sequences – the dark matters in the potential proteome universe.

In addition to per-token log-likelihood and sequence identity, a variety of existing protein predictors may be used to prioritizing the generated protein candidates from different perspectives, such as protein structure[22] and enzyme commission (EC) number.[23] Johnson et al. also incorporated multiple sequence-based and structure-based tools (e.g., Rosetta,[24] ESM-v1,[25] CARP-640M,[26] ProteinMPNN,[22] MIF-ST,[27] and ESM-IF[28]) for predicting whether the generated enzymes possess *in vitro* activity. [29] However, using these existing predictors suffer from multiple limitations. First, none of these predictors directly reports the likelihood of the desired functions. Second, it requires non-trivial efforts in benchmarking and programming environment setup. Third, these predictors may not be applicable to different types of proteins (e.g., EC number prediction only applies to enzymes).

To address the limitations of current prioritization strategies, here we coupled ProtGPT2 with a convolutional neural network (CNN)-based protein discriminator. Positive protein data were used to finetune the protein generator, ProtGPT2; while both positive and negative data were used to train the discriminator, which was then used to prioritize the generated sequence candidates, allowing a one-stop solution to DNPD for different types of proteins without the dependencies on external tools. A pre-trained LLM, ProtTrans, was used for protein feature extraction for the discriminator. Such LLM-based contextualized embedding technique has been proven to outperform traditional protein features such as one-hot encoding.[30-32] The efficacy of the framework was then demonstrated by two distinct types of sequences: AMPs as representatives of short sequences lacking homology, and malate dehydrogenases (MDHs) as representatives of longer sequences with inferable homology. In addition, the prompts length and size of finetuning set were investigated to fully elucidate the potential of this framework. Altogether, our results positioned this framework as a powerful and efficient tool to advance DNPD.

MATERIALS AND METHODS

Data preparation

We downloaded 3,011 antibacterial peptides from SATPdb database[33] (http://crdd.osdd.net/raghava/satpdb/, accessed on 8/24/2022) as positive set. We filtered this peptide set to only include those composed of canonical amino acids and that were less than or equal to 50 residues in length, resulting in 2,879 peptides, which we refer to as the AMP set. For the negative set, we downloaded 13,225 short (less than or equal to 50-residue-long) protein sequences/peptides from UniProt[34] (https://www.uniprot.org/, accessed on 8/16/2022). We removed sequences containing certain keywords ("antimic*", "antibac*", "*toxic*", "defensive") from the UniProt description. This negative set curation is consistent with other relevant methods.[35] To create a balanced negative set and to avoid bias in sequence length distribution, we randomly sampled from this UniProt pool strictly following the AMP set's length distribution. For example, if there were 179 AMPs that are 21-residue-long, then we sampled exactly 179 sequences that are 21-residue-long. In total, we collected 2,879 non-AMPs as negative set.

We gathered a positive set of 16,706 MDH sequences from the training set of Repecka et al.'s study.[21] For negative set, we searched "enzyme" on UniProt (https://www.uniprot.org/) and downloaded all reviewed sequences (n = 568,002, accessed on 8/16/2022). To later evaluate how well our model discriminates MDH from functionally close enzymes, we used the validation

Table 1. Summary of the MDH sequence generation.

Group name	Group meaning	Template	Starter seq. length	No. generated seq.
random	randomly generated according to the amino acid occurrence distribution of all MDH	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000
noft	generated by non-finetuned ProtGPT2	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000
cluster765	finetuned by *cluster765*	representative seq. of *cluster765*	0	1000
			10	1000
			50	1000
			100	1000
cluster2029	finetuned by *cluster2029*	representative seq. of *cluster2029*	0	1000
			10	1000
			50	1000
			100	1000
cluster5987	finetuned by *cluster5987*	representative seq. of *cluster5987*	0	1000
			10	1000
			50	1000
			100	1000
cluster7477	finetuned by *cluster7477*	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000

set (n = 213) from Repecka et al. as positive set; we also identified 5,766 EC1.1.1.X sequences (first three enzyme commission [EC][36] levels are 1.1.1 and the last level is not 37) and 543 EC1.1.X.X sequences (first three EC levels are 1.1 and the third level is not 1) as negative set. We then narrowed these sequences to those a) whose EC numbers are not 1.1.1.37 (the EC number of MDH); b) whose sequence lengths are within the range of the positive set's (64-505); and c) that are not functionally close (EC1.1.1.X or EC1.1.X.X), resulting in 206,596 non-MDH sequences as negative set.

To compile a balanced negative set without bias in sequence length distribution, we enumerated the number of sequences by an interval of 50 and sampled that amount of non-MDH sequences for that interval. For example, if there were 39 MDH sequences that are 110-160 residues long, then we sampled 39 non-MDH sequences of that length range. As a result, we collected 16,706 non-MDH sequences as negative set.

Since the 16,706 MDH sequences may vary a lot (with a length range of [64-505]), to ensure consistency during the finetuning process, we clustered the MDH sequences into multiple clusters using cd-hit.[37] To reduce the number of resulting clusters, we used the lowest allowed sequence identity threshold (0.4) and word length (2) for clustering. A total of 82 clusters were generated and the majority of these clusters are singletons or very small. The four largest clusters constituted 97.3% (n = 16,258) of all MDH sequences. We termed these four clusters as *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*, with the number inside the name indicating the number of sequences in that cluster (e.g., *cluster765* contain 765 sequences). During clustering, a representative sequence is generated for each cluster, which is essentially the first sequence in sequential order that does not belong to any previous formed clusters.

Generator finetuning and sample generation

For the AMP generation task, we finetuned ProtGPT2[11] (https://huggingface.co/nferruz/ProtGPT2) with default parameters on the AMP set. We then generated 3,000 sequences using text generation pipelines from Python (v. 3.8.15) transformers package (v. 4.26.0.dev0)[38] (https://huggingface.co/docs/transformers/index) with maximum lengths being 20. Note that the generated sequences may exceed this maximum length. As negative control for the finetuning process, we also generated 3,000 sequences using the non-finetuned ProtGPT2 model.

For the MDH generation task, we finetuned the ProtGPT2 model using four separate datasets: *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*, resulting in four finetuned models. To generate new sequences, we supplied each finetuned model with a starter sequence of varying length, ranging from 0-100 residues, taken from the representative sequence of the corresponding finetuning set. The maximal length for generation was set to be 400. The results of this sample generation are summarized in Table 1 below. All finetuning tasks were conducted on NVIDIA A100 (80G) graphical processing unit (GPU).

Discriminator training

For the AMP discriminator, we split the balanced AMP/non-AMP set (n = 2,879+2,879) into training, validation, and testing set in 8:1:1 ratio. From the MDH/non-MDH balanced set (n = 16,706 + 16,706), we randomly sampled 4,500 sequences for training (13.47%), 500 sequences for validation (1.50%),

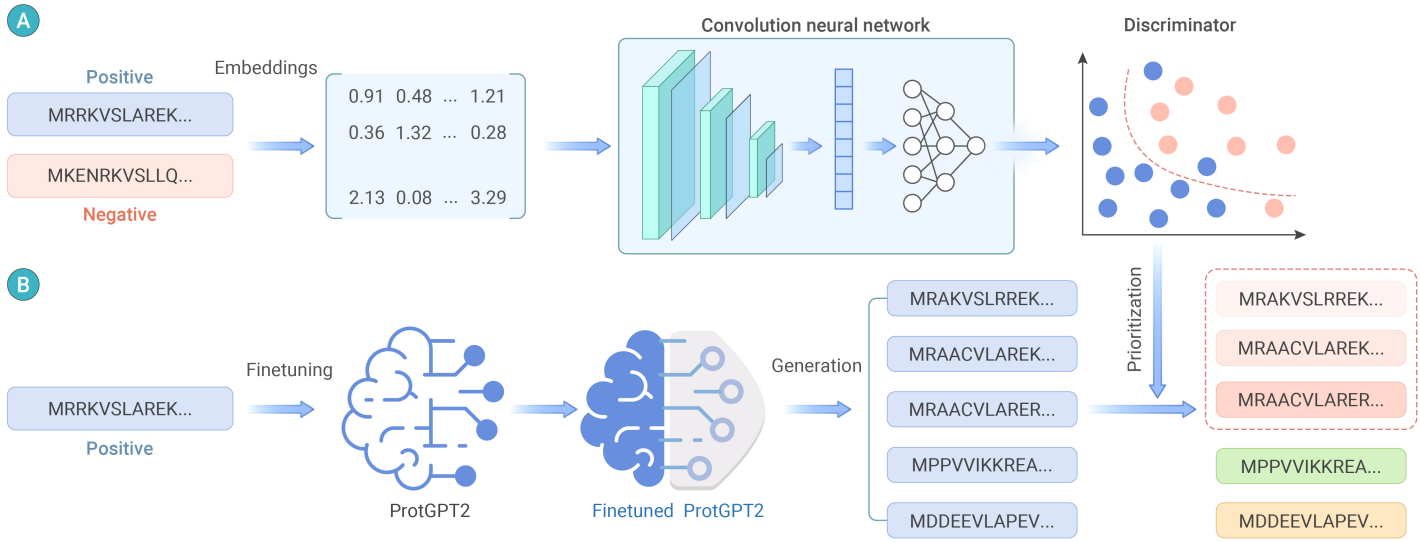


Figure 1. Discriminator (A) and generator (B) architecture In panel A, positive and negative sequences are embedded using ProtTrans (each residue is converted to a 1,024-value long vector). The embedded inputs are fed to a neural network with three convolution layers and three max-pooling layers, followed by a fully connected neural network. Meanwhile, in panel B, the positive sequences are used to finetune ProtGPT2, which is then used to generate positive candidate sequences. In the end, the discriminator is used to screen the most likely positive sequence candidates.

and the remaining 28,412 sequences were used for testing (85.03%). We performed discriminator training, hyperparameter tuning, and performance evaluation on training set, validation set, and testing set, respectively.

To extract features from the protein sequences, we utilized the ProtT5-XL-UniRef50 model, which is part of the ProtTrans project.[39] We chose ProtT5-XL-UniRef50 among others in ProtTrans because it achieved the best performance most often in different predictive tasks.[39] We used the transformers package (v. 4.26.0.dev0) in Python (v. 3.8.15) to implement the embedding, with maximum lengths of 50 for the AMP task and 505 for the MDH task.

For the discriminator architecture, we used Keras package (v. 2.10.0)[40] to implement a convolutional neural network (CNN) followed by a fully connected neural network (FCNN) (Figure 1). Each convolution layer was coupled with a maximum-pooling layer, and the last maximum-pooling layer was flattened, which was fed to the FCNN. Recticfied linear unit[41] was used as activation function for all layers except for the output layer, where the activation function was sigmoid function[42] instead. We then used Adam[43] as the optimizer, binary cross-entropy (Eqn. 1) as the loss function, and accuracy (Eqn. 2) as the evaluation metric. We set maximum number of training epochs to be 100 with early stopping (patience=5). We adopted the following hyperparameter setup after tuning: three 3×3 convolution filter, each followed by 2×2 max-pooling layer; a flatten layer; three fully connected layers (with 1024, 512, 256 nodes, respectively) and one output layer. Tunable hyperparameters include number of filters in CNN, kernel size in CNN, maximum-pooling size, number of dense layers in FCNN, number of nodes in each FCNN layer, batch size, and learning rate of optimizer. Embedding and training were carried out on a NVIDIA A100 (80G) graphical processing unit.

$$\text{binary cross\_entropy} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot \log(1-\hat{y}_i)) \quad (1)$$

where N is the number of samples, y_i and $\hat{y}_i$ are the *i*-th actual and predicted label, respectively.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where *TP*, *TN*, *FP*, and *FN* are true positive, true negative, false positive, and false negative, respectively.

Bioinformatics analysis

We performed Pfam domain[44] scanning on all generated MDH sequences using HMMER package[45] (v. 3.3.2). Pfam profiles were built using hmmerbuild command with Pfam seed alignment flatfile and Pfam profile flatfile (downloaded from http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.seed.gz and http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.hmm.gz, respectively. accessed on 11/10/2022). The E-value threshold for Pfam domain scanning was set as 0.001. We considered a sequence to have MDH's signature domain if at least one "lactate/malate dehydrogenase" or "Malate/L-lactate dehydrogenase" domain was identified.

We evaluated the sequence redundancy within *cluster765*, *cluster2029*, *cluster5987*, *cluster7477*, as well as the generated MDH sequences using cd-hit (parameters set as default). We defined "ratio of non-redundant sequences" as the number of non-redundant sequences determined by cd-hit divided by the number of input sequences.

For every generated MDH sequence, we performed pairwise alignment using BLAST standalone package[46] (v. 2.13.0+) against all known MDH sequences (n = 16,706) and identified the "best hit" (highest bit score) sequence. We recorded the identity between query (generated) sequences and their corresponding best hit sequence. Based on these best hit sequences' identity and ungapped alignment length, we utilized HFSP curve (homology-derived functional similarity of proteins,[47] HFSP=0) to visualize whether a generated sequence share the same function as its best hit known MDH.

Sequence space visualization using t-SNE

We randomly selected 100 known MDHs from *cluster2029*, 100 MDH candidates generated by *cluster2029*-finetuned ProtGPT2 that are prioritized (non-redundant and discriminator prediction>0.95), 100 EC1.1.1.X and 100 EC1.1.X.X sequences for visualization of sequence spatial distribution with t-distributed stochastic neighbor embedding (t-SNE) algorithm.[48] Each of the selected sequences was embedded by the ProtT5-XL-UniRef50 model[39] as an *L*×1024 matrix (where *L* is the length of the protein), which was then flattened into a *L*×1024-long vector as protein-level features. The t-SNE transformation from the embedded features to two-dimensional space was conducted using Python scikit-learn package (v1.1.2).[49]

Mapping conservation to enzyme functionally important sites

Finetuned-generated MDH candidates that are a) non-redundant and b) with discriminator prediction>0.95 were selected. We used a known MDH sequence (UniProt ID: P11708) with determined protein structure (protein data bank [PDB][50] ID: 4MDH) as the reference sequence. The reference sequence was BLASTed against the selected MDH candidates. From the selected MDH candidates with significant hit, 499 were randomly sampled and then subjected to multiple sequence alignment (MSA) along with P11708

using MAFFT[51] program. The MSA-derived entropy scores were computed using the msa package[52] in R. The conservation scores were calculated as the maximal entropy minus the actual entropy at a given position, so that higher score indicates higher conservation. Annotation of functionally important sites (substrate binding, NAD binding, and sulfate binding) of 4MDH were obtained from PDB. Conservation scores were mapped to residue positions in 4MDH. Visualization of 4MDH was performed using PyMOL.[53]

Prediction of binding affinity between MDH and malate

To evaluate the binding affinity between MDH sequences and the substrate, malate, we randomly sampled 50 generated MDHs with discriminator predicted scores > 0.9, as well as 50 natural MDHs for comparison. The natural MDHs were sampled with a constraint that the length distribution should be the same as the generated MDHs'. The binding affinity between MDH and malate was predicted using two programs: PSICHIC[54] and SMINA.[55] PSICHI is a machine learning-based predictor with protein sequence and ligand as input; while SMINA is a docking program with protein structure and ligand as input. ESMFold[56] was used to predict the structures of the selected MDH sequences.

Statistical analysis and visualization

Statistical analysis and visualization were conducted in R (v. 4.2.2) with ggplot2 package (v. 3.4.0). Calculation for area under the receiver operating characteristic curve (ROC-AUC) was done with pROC package[57] (v. 1.18.0).

Experimental validation

Experimental validation on AMP and MDH candidates is described in Supplementary Materials.

RESULTS

Overview of the DNPD framework

We developed a convenient one-stop solution (Figure 1) for general-purpose protein design by integrating a state-of-the-art protein generative model, ProtGPT2[11] with a custom-built protein discriminator. The ProtGPT2 model, a large-scale language model with 738 million parameters, serves as the foundation for generating new protein sequences. The protein discriminator, consisting of a convolutional neural network followed by a fully connected neural network (Figure 1A), is used to assess and prioritize the generated sequences. Protein sequence features are extracted using ProtT5-XL-UniRef50, a model-based embedding framework in ProtTrans collection, which comprises of 3 billion parameters.[39] The generative model is finetuned with a positive set of protein sequences, while the discriminator is trained on a balanced dataset containing both positive and negative protein sequences. As proof of concepts, we applied this framework to the design of two distinct protein classes: antibacterial peptides (AMP) and malate dehydrogenases (MDH; EC1.1.1.37).

Discriminators accurately classifies AMP/non-AMP and MDH/non-MDH sequences

For the AMP/non-AMP discriminator, the AMP and non-AMP sequences were obtained from SATPdb database[33] and UniProt,[34] respectively. Short peptides from UniProt that are shorter than 50 residues and without certain antimicrobial-relevant keywords were selected as non-AMPs.[35] Similarly, for the MDH/non-MDH discriminator, we acquired MDH sequences from Repecka et al.[21] and non-MDH sequences randomly sampled from UniProt.[34] To ensure unbiased results, all training, validation, and test sets were carefully balanced by randomly sampling negative sets to mirror the sequence length distribution in positive sets. The discriminators achieved high accuracies and converged rapidly (< 100 epochs), demonstrating the effectiveness of transfer learning-based embedding techniques. Specifically, the AMP/non-AMP discriminator achieved an accuracy of 0.81 on a balanced testing set of 576 samples (Figure 2A), while the MDH/non-MDH discriminator attained an accuracy of 0.99 on a larger balanced test set of 28,412 samples (Figure 2A). It is important to note that numerous AMP/non-AMP discriminators have been developed by other researchers.[58] However, our framework is meant for de novo design for peptides/proteins in general, instead of building a discriminator for a specific task such as classifying AMP.

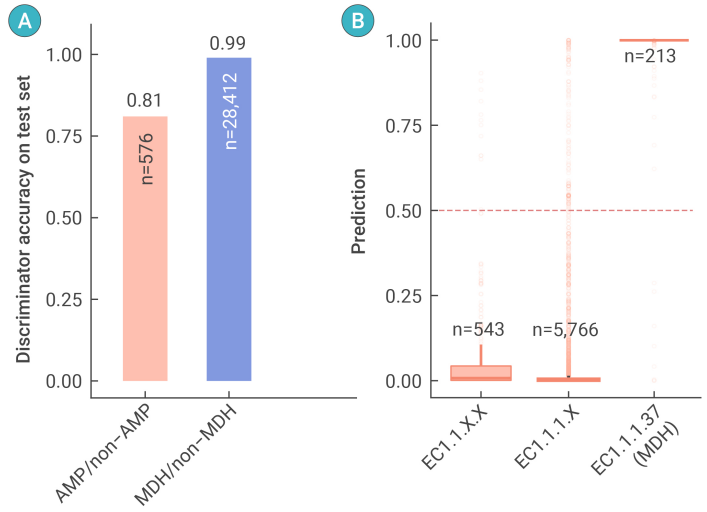


Figure 2. Accuracies (A) and resolutions (B) of the discriminators In panel A, numbers denoted inside the bars are size of testing set (both balanced). Panel B shows MDH/non-MDH discriminator's predictions on MDHs and functionally close enzymes: "EC1.1.1.X" are enzymes with the first three EC digits being 1.1.1. (excluding EC1.1.1.37); likewise, "EC1.1.X.X" are enzymes with the first two EC digits being 1.1. (excluding EC1.1.1.X). Numbers above the boxes are data size, and all data are not included in training set. AUCs for EC1.1.X.X vs. MDH and EC1.1.1.X vs. MDH are both 0.99.

To further evaluate the discriminators' capability to differentiate between functionally close enzymes, we assembled additional datasets comprising EC1.1.1.X enzymes (n = 5,766, excluding EC1.1.1.37) and EC1.1.X.X enzymes (n = 543, excluding EC1.1.1.X), as well as an additional reserved set of 213 MDHs for further examination. All of these sequences were unseen during training. The MDH/non-MDH discriminator achieved a 0.99 ROC-AUC for both EC1.1.1.X vs. MDHs and EC1.1.X.X vs. MDHs (Figure 2B), highlighting its exceptional ability in differentiating between functionally similar enzymes. Overall, the results demonstrated the effectiveness of the discriminators in accurately distinguishing protein sequences of interest.

Finetuning increases the likelihood of the generated sequences being AMP or MDH

Finetuning essentially involves adapting a pre-trained LM to a custom input dataset, enabling the updated LM's to generate content more closely aligned with the input's style.[59] We finetuned ProtGPT2 with 2,879 known AMPs and subsequently generated 3,000 AMP candidates using the finetuned model. During generation, the "maximal length" was set to 20 and no starter sequence was provided, however, note that the actual generated sequences may exceed the set "maximal length". For comparison, we created an equal amount of a) random sequences, i.e., randomly constructed sequences following the length distribution as natural AMPs; b) "AA-imitating", i.e., following natural AMPs' amino acid and length distribution; c) ProtGPT2-random sequences, i.e., sequences generated by the non-finetuned ProtGPT2. As expected, the finetuned-generated sequences are more likely to be actual AMPs than "AA-mimicking" and "ProtGPT2-random" (prediction median 0.87 vs. 0.37 and 0.01, Wilcoxon signed rank test p-value<2.2e-16; Figure 3A). Surprisingly, we noticed that the "AA-mimicking" group also reached moderately high prediction (median = 0.37; Figure 3A). This can be attributed to two reasons—a) there lacks gold standard negative set and our positive set is relatively small (excluding anti-yeast, anti-viral, etc.); and b) cationic charge and hydrophobicity are the dominant determinants of antimicrobial effects.[60-62] Since these "AA-mimicking" peptides closely mimics the amino acid profile of natural AMPs, they inherited the cationic charge and hydrophobicity and are thus more likely to possess antimicrobial effects.

To explore the impact of data size and sequence redundancy on finetuning results for MDHs, we clustered the 16,706 known MDHs and selected the four largest clusters (termed *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*) as finetuning sets, comprising 765, 2029, 5987, and 7477

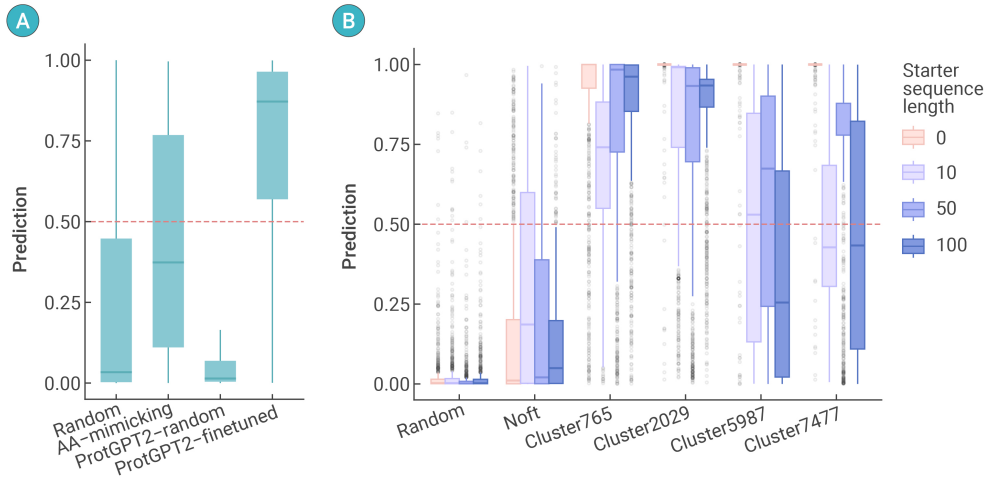


Figure 3. Effects of generator finetuning on AMP (A) and MDH (B) likelihood of the generated sequences Panel A shows the discriminator's predictions on four groups of peptide sequences. Group "random" means randomly constructed peptides; group "AA-mimicking" means peptides constructed following the amino acid and length distribution of the actual AMPs; group "ProtGPT2-random" means peptides generated by ProtGPT2 without any finetuning; group "ProtGPT2-finetuned" means peptides generated by ProtGPT2 that was finetuned with actual AMPs. In panel B, group "random" means random sequences following amino acid distributions MDHs; group "noft" in panel B means candidates generated by non-finetuned ProtGPT2. Other groups in panel B (*cluster765*, *cluster2029*, *cluster5987* and cluster7377) indicates sequences generated by ProtGPT2 finetuned by the respective sequence cluster (the numbers indicate corresponding cluster size).

sequences, respectively. As controls, we generated two sets of sequences: a) random sequences that followed the amino acid and length distribution of the known MDHs; and b) sequences generated by the non-finetuned model. Discriminator predictions revealed that sequences generated by the finetuned models were significantly more likely to be MDHs than random and non-finetuned model-generated sequences (prediction median 0.91 vs. 0.003 and 0.91 vs. 0.033, respectively; Wilcoxon signed rank test p-value < 2.2e-16 for both comparison; Figure 3B). Notably, we found that even a relatively small number of sequences (~700, in the case of cluster765) were sufficient to finetune ProtGPT2 for generating high-quality MDH sequences. Importantly, the discriminator's predicted scores do not correlate with ProtGPT2's perplexity during generation (Pearson correlation= -0.01). Considering the aforementioned limitations of perplexity, we further demonstrated the superiority of our discriminator in candidate prioritization.

Note that users can choose to provide a leading sequence to prompt ProtGPT2 during sequence generation, which we refer to as "starter sequence". To investigate whether such a starter sequence would guide the generation towards our desired functions, we incorporated starter sequences of variable lengths. We supplied the first 0 (i.e., none), 10, 50, or 100 residues of a representative sequence (the representative sequence in the corresponding cluster; see Materials and Methods) as a starter sequence during the generation process. As a result, providing a starter sequence did not improve the quality of the generated sequences, as candidates generated without starter sequences generally had higher discriminator predicted scores (Figure 3B). Notably, in cluster5987 and cluster7477, employing a starter sequence adversely impacted prediction accuracy. This suggests that, for a diverse training set, the inclusion of a starter sequence might impose unnecessary constraints on sequence generation.

We observed that most of the generated sequences were non-redundant identified by cd-hit,[37] except for those generated by the *cluster2029*-finetuned model (ratio of non-redundant sequences =0.31 vs. 0.67-0.95 in other generated groups), likely due to the low non-redundancy within the cluster itself (ratio of non-redundant sequences =0.08 vs. 0.28-0.33 in other clusters; Figure 4A). The effect of supplying a starter sequence on the redundancy of generated sequences was not conclusive (Figure S1).

Prioritized generated AMP candidates are endowed with antibacterial activity

With the generated AMP candidates and their corresponding discriminator predictions available, we randomly selected 24 AMP candidates with predicted scores greater than 0.95; meanwhile, 10 random peptides were generated as control (Table S1). The amino acid and peptide length distributions of the random peptides are the same as the natural AMPs'. These peptides were tested against six different bacterial strains, including four Gram-positive strains (*S. aureus*, *K. rhizophila*, *B. pumilus*, and *B. subtilis*) and two Gram-negative strains (*E. coli* GDMCC 1.335 and DH5α). Out of the 24 generated AMP candidates, six completely inhibited the growth of at least one bacterial strain (Figure S2). Notably, five of these candidates displayed broad-spectrum activity against all six bacterial strains. In comparison, only one of the randomly selected peptides exhibited activity against some bacterial strains, highlighting the presence of many unknown AMPs in nature, as reported in previous studies.[63,64] To accurately assess the potency of these candidates, five broad-spectrum AMPs were purified to 90% purity and their minimum inhibition concentrations (MICs) were determined by a growth-based assay. Four of these candidates displayed remarkably low MICs (with the lowest at 1.875 μM, Table 2), highlighting their potential for therapeutic and industrial applications.

Most finetuned-generated MDH candidates harbor MDH domains

A domain is a structural unit that carries a certain protein function. The function of MDHs is characterized by a "lactate/malate dehydrogenase" domain or a "Malate/L-lactate dehydrogenase" domain (here we collectively denote as "MDH domain") catalogued in Pfam database.[44] Thus, bearing such signature domain serves as a good indicator of generation quality in addition to our discriminator. We found that not a single random or non-finetuned-generated sequence has the MDH domain (Figure S3), suggesting that this domain is extremely unlikely to occur by chance. In contrast, the vast majority (94.54%) of finetuned-generated sequences have MDH domains, increasing the confidence in their potential to possess the desired MDH function. Besides, the discriminator's predictions on sequences with MDH domains are significantly higher than those without MDH domains (prediction median 0.95 vs. 0.02, Wilcoxon signed rank test p-value < 2.2e-16; Figure 4B), assuring that the discriminator can well capture the sequence-function relationship.

Homology inference suggests MDH function in most finetuned-generated MDH candidates

The MDH candidates generated through finetuning exhibit high sequence identity (median = 83.65%) with their best hit known MDH (Figure 4C). Meanwhile, we believe that some high identity candidates may not actually be MDH (Figure 4D, bottom right corner) according to our discriminator. In contrast, some candidates with relatively low best hit identities (50-60%) exhibit high predicted scores, enabling researchers to select MDH candidates with a relatively high degree of novelty. Next, to further validate the utility of the discriminator in light of homology, we selected a set of generated candidates highly likely to be MDH, i.e., prediction > 0.95, defined as *prioritized candidates*. We applied HFSP (homology-derived functional similarity of proteins)[47] to the *prioritized candidates*. HFSP assesses whether a pair of proteins share the same function based on ungapped alignment length and alignment identity. Although resolution of HFSP may not be able to reach the fourth level of EC number,[47] it is still a useful baseline for homology reference. Notably, HFSP predicted that all prioritized generated candidates have the same function as their corresponding best hit sequence (Figure 4F), i.e., a known MDH. These results reinforced our confidence in the quality and functionality of the MDH sequences generated through finetuning, as supported by homology analysis.

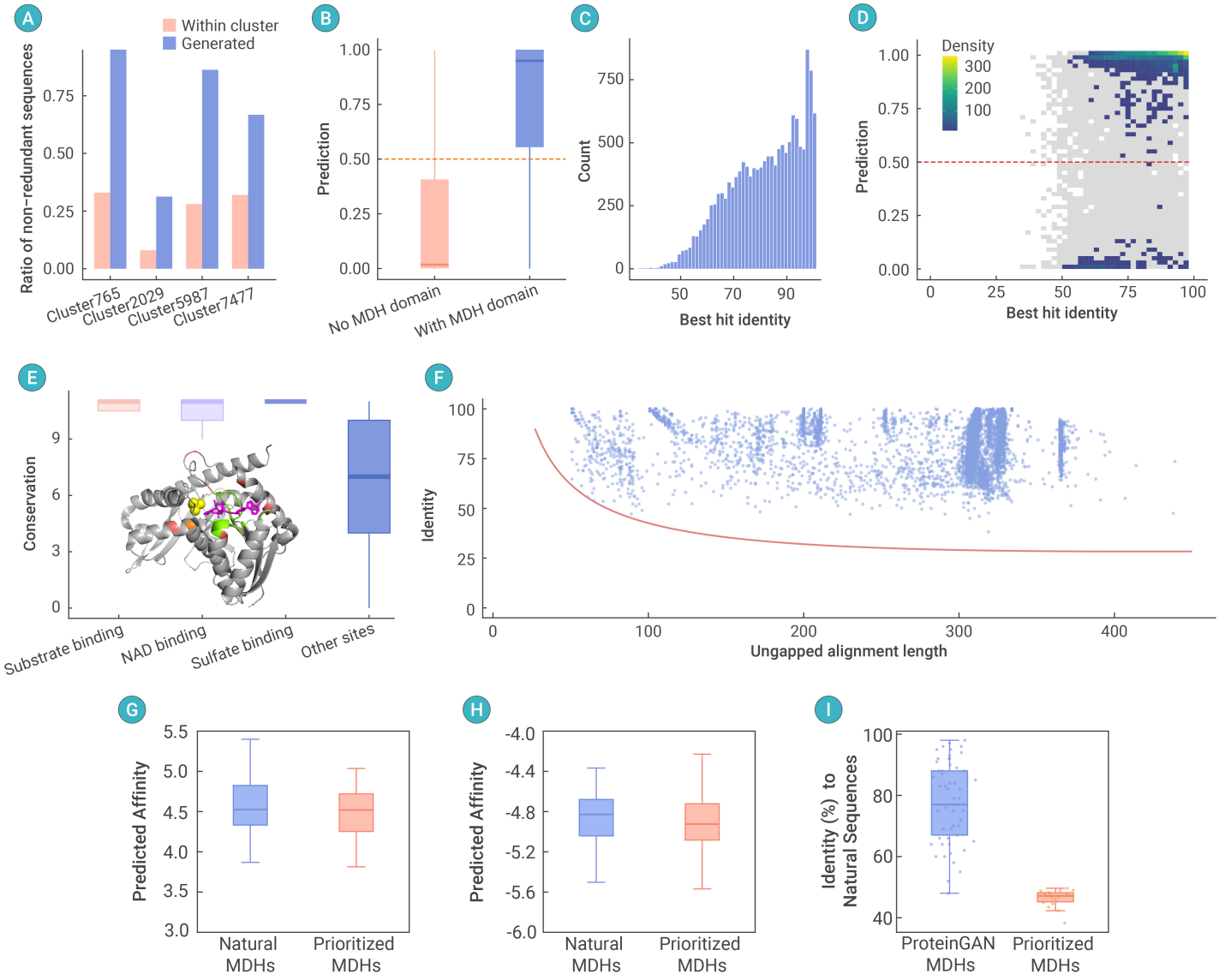


Figure 4. Computational analysis and evidence for the functionality of the prioritized MDH candidates Panel A shows the ratios of non-redundancy within each cluster (*cluster765*, *cluster2029*, *cluster5987* and cluster7377) and among the generated sequences finetuned by the corresponding cluster. Panel B shows discriminator predictions on generated MDH candidates with and without MDH domains. Panel C shows the histogram of best hit identity of the generated MDH candidates. Panel D shows discriminator's prediction vs. best hit identity (presented as data point density, grey grids contain <10 data points) for the finetuned-generated MDH candidates. Panel E shows the conservation along the MDH candidates with respect to functionally important sites. Position-specific conservation scores were derived from multiple sequence alignment (MSA) among 500 sequences (a known MDH sequence [UniProt ID: P11708] and 499 randomly sampled finetuned-generated MDH candidates with discriminator prediction>0.95). The determined protein structure (PDB ID: 4MDH) of P11708 was added to the MSA to map the protein positions to functionally important sites. The structure image depicts the structure of 4MDH with ligands sulfate (yellow spheres) and NAD (purple licorie) highlighted. Resides highlighted in pink, green, and orange indicate substrate binding, NAD binding, and sulfate binding sites, respectively. The boxplot summarizes the conservation scores at different sites. Panel F shows the best hit's identities and ungapped alignment lengths of generated sequences (blue dots) with high predicted scores (>0.95). Red curve in panel E indicates HFSP curve: sequence pairs above and below the curve are predicted by HFSP as having the same and different enzyme functions, respectively. Panel G-H shows the MDH-malate binding affinity predictions by PSICHIC and SMINA, respectively. Panel I shows the novelty (i.e., sequence identity to the closest natural sequence) of ProteinGAN-generated MDHs and our prioritized MDHs.

Table 2. MIC of the identified broad-spectrum AMPs.

Peptides ID	sequence	length	prediction	mass	MIC (μM)					
					S. aureus GMDCC1.221	*B. pumilus* GMDCC1.225	*B. subtilis* GDMCC1.222	*K. rhizophila* GDMCC1.226	*E. Coli* GDMCC1.355	*E. Coli* DH5α
AMP3	GLFKIFKKSVKHACKKLK	18	0.97	2103.7	15	3.75	3.75	15	7.5	3.75
AMP6	KYLRRKGGRKWVKNAFRNLM	20	0.97	2522.1	60	15	60	120	120	120
AMP13	YRGIRGKGWWKAAKAALKAVSAAAKH	26	0.99	2796.3	1.875	1.875	1.875	3.75	15	7.5
AMP23	KIKEKLRQKLGNIMEKHLHKLAHRILLKK	29	0.98	3545.4	3.75	15	7.5	60	30	15

Prioritized MDH candidates are close to known MDHs in sequence space

To visualize spatial relationships between sequence-derived features of the known MDHs and the discriminator-prioritized MDH candidates, we first embedded the protein sequences into L×1024-dimensional vectors (where L

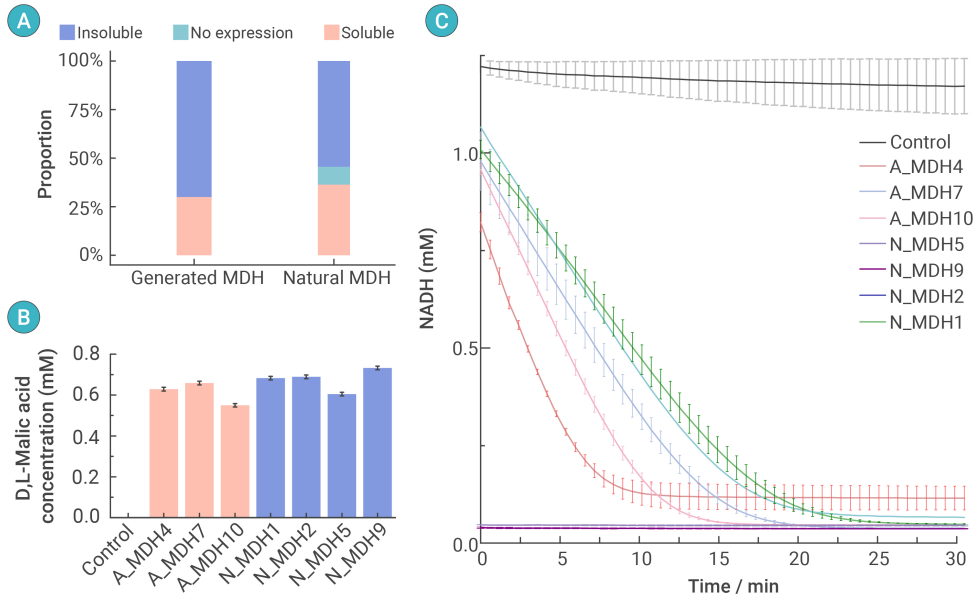


Figure 5. Experimental validation on the generated and prioritized MDH candidates Ten prioritized (prediction > 0.9) and novel (< 50% sequence identity to any natural MDHs) were randomly selected. Ten natural MDHs were also randomly selected as positive control. Samples without recombinant enzymes served as negative control. Results showed that 3/10 generated candidates and 4/10 positive controls were expressed and soluble (no significant difference; proportion test p-value = 1), as displayed in panel A. Panel B shows the malic acid concentration transformed from oxaloacetic acid. Panel C shows the consumption of NADH over time. Prefix letter A and N indicates the artificially generated group and the natural (positive control) group, respectively. For example, "A_MDH4" indicates the fourth generated MDH candidate and "N_MDH1" indicates the first positive control.

is the length of a protein) using ProtTrans ProtT5-XL-UniRef50, we than used t-SNE algorithm[48] to transform the vectors into a two-dimensional space. Here, we randomly selected 100 known MDHs and 100 *prioritized candidates* to be visualized. To avoid bias caused by redundant sequences, we further limited the *prioritized candidates* to be non-redundant with each other determined by cd-hit[37] during the random selection. Functionally similar enzymes (EC1.1.1.X and EC1.1.X.X) were also added into the comparison as reference. The resulting visualization (Figure S4) showed that known MDHs and the *prioritized candidates* tend to cluster together, while functionally similar enzymes barely locate in the space between the known MDHs and the prioritized MDH candidates. This observation suggests that, in terms of sequence context, the *prioritized candidates* are in line with the known MDHs and are distinct from functionally similar enzymes.

Prioritized MDH candidates are conserved at important functional sites

We randomly selected 499 *prioritized candidates* to perform multiple sequence alignment (MSA) with a known MDH with structural information (UniProt ID: P11708; PDB ID: 4MDH). From the MSA, we derived position-specific conservation scores and mapped them to 4MDH's structural annotations. We found that the conservation scores at substrate binding sites, NAD binding sites, and sulfate binding sites were significantly higher than those at other sites (11 vs. 7, t-test p-value < 0.00025 for all three comparisons; Figure 4E). These results suggest that our framework has successfully learned the sequence patterns required for MDH function and that the resulting MDH candidates are likely to perform this function.

Prioritized MDH candidates have similar substrate affinity as natural MDHs

We randomly selected 50 generated candidates with predicted scores > 0.9 and 50 natural MDHs. We then ran two different protein-ligand affinity prediction programs between the MDHs and malate. We found that, forboth programs, the distributions of the predicted affinities between the generated MDHs and the natural MDHs are the same (Kolmogorov-Smirnov test, p-value > 0.05). These results provide additional evidence for the functionality of the candidate sequences designed by our framework.

Prioritized MDH candidates are functional in vitro

We identified 18 prioritized (predicted score > 0.9 and identity < 50% to any natural MDHs) MDHs from the generated candidates. These prioritized sequences are significantly more novel than the ones generated by ProteinGAN (t-test p-value < 0.05, Figure 4H). For experimental validation, we further randomly downsampled the prioritized sequences to ten sequences. As positive controls, we randomly selected 10 natural MDHs from the training set (n=16,705) of ProteinGAN.[21] Information of these sequences is summarized in Table S2. We expressed these proteins in *E. coli* BL21 (DE3) system (Figure S5). We observed that only 3/10 MDH candidates and 4/10 positive controls were soluble (Figure 5A). Activity assays showed that all the soluble MDH candidates and the positive controls are functional in vitro (Figures 5B-C), showcasing the utility of our framework in designing functional and novel enzymes. The concurrent failed expression in both de novo designed MDHs and natural MDHs may be attributed to multiple factors, such as codon optimization, chaperones, and posttranslational modifications.[65]

DISCUSSION

AMPs offer hopes in addressing the global antibiotic-resistance crisis[66] due to their advantages over small molecule antibiotics, e.g., modulation of immune response, broad-spectrum activity, multiple mechanisms of action, and diverse source organisms from all domains of life.[67] Numerous attempts have been made to predict and *de novo* design novel AMPs.[68-70] Recently, there are four deep generative language model-based methods for AMP design with experimental validation.[64,71-73] The first three methods involved an RNN-based generator and a predictor, both trained on known AMPs. In addition to the predictor, a variety of physicochemical properties were also employed to facilitate candidate filtering. The fourth method[73] is more complex—an autoencoder generator was trained on all known short peptide sequences to better reflect peptide properties and enable high diversity. Meanwhile, multiple attribute classifiers (such as antimicrobial activity, toxicity, sequence similarity, and physicochemical properties) were trained on the latent space of the autoencoder, and the generation process was guided by these attribute classifiers altogether for desired peptide properties, and the attribute classifiers and a series of external predictors were used for candidate screening. All of these language model-based methods, as well as GAN-based methods,[74] have successfully identified experimentally valid novel AMPs, some of which are broad-spectrum. However, performance comparison across these methods is difficult due to variations in data size, time consumption for training, threshold of "inhibitory effect", definition of "broad-spectrum", bacterial strains tested, peptide purity, and sequence novelty. From a usage convenience standpoint, our framework has two major advantages over other language model-based methods. First, it does not need to train a generator from scratch, as the existing generator (ProtGPT2) assumably performs better due to the tremendously larger training protein dataset. Second, the candidate prioritization is straightforward as it relies on only one discriminator, instead of multiple classifiers or additional calculation of physicochemical properties.

Enzyme engineering has become an essential aspect of metabolic engineering and synthetic biology.[75,76] When incorporating a heterogenous enzyme into an engineering chassis, it is often necessary to adjust the enzyme properties to better fit the working condition. Generating artificial protein sequences offers more options for selecting the ideal enzyme version. For example, a variety of tools can be applied to screen for more suitable solubility,[77,78] catalytic capacity,[79,80] and protein-protein interaction.[81,82] Many

attempts have been made for enzyme DNPD.[83,84] One of the most successful non-LLM frameworks is ProteinGAN,[21] a GAN-based generator. As a showcase, ProteinGAN was trained on 16,706 MDHs and produced 55 candidates for testing, of which 13 candidates experimentally displayed MDH catalytic activity. Notably, GAN is suitable for continuous data (such as images), but not for discrete data (such as text), because GAN relies on gradient-based optimization and there is no smooth gradient between distinct tokens.[85] When applied in text (such as protein) generation tasks, GAN also suffers from training instability,[86] ignorance on sequential dependencies,[87] mode collapse,[88] and result evaluation.[89]

Compared to ProteinGAN, our framework has several advantages. First, our discriminator was trained on much less protein sequences than ProteinGAN (n = 2,029 vs. n = 16,706) without compromising the accuracy (0.99 accuracy on a much larger testing set: n = 28,412). Second, it can distinguish MDHs from functionally similar enzymes at a high resolution, surpassing the limitations of traditional homology-based methods.[47] The candidate screening process is therefore remarkably simplified and improved, eliminating the need for additional predictors and replacing sequence identity-based criteria with a straightforward sequence-function scoring scheme. With that said, it allows us to select both conservative (high sequence identity) and novel (low sequence identity) candidates while maintaining high functional fidelity (high discriminator scoring). Last but not least, our framework is time-efficient. The time consumption for the MDH task (in the case of the *cluster2029* for example) is dissected as follows: ~10 min generator finetuning (n = 2,029), ~1 h embedding the training/validation sets (n = 4,500 + 500), 2 min discriminator training/validation (n = 4,500 + 500), ~4 h generation (1,000 sequences), ~20 min embedding the generated sequences (n = 1,000) , and ~1 min prediction. The MDH task can be completed in 6 hours (without parallel computing), which is a small fraction of the time it took for ProteinGAN (9 days). Designing novel MDHs was also attempted in Madani et al.'s work on ProGen.[12] In their work, the aforementioned per-token log-likelihood scoring, as well as ProteinGAN's discriminator were used for the prioritization on the generated MDH candidates, achieving 0.94 and 0.87 ROC-AUC, respectively, on 56 samples obtained from the ProteinGAN paper.[21] In comparison, our discriminator achieved higher ROC-AUC (0.99) on a much larger test set (n = 28,412), demonstrating the accuracy of our discriminator. Notably, our framework is capable of designing far more enzymes than MDH alone. For additional demonstration, we successfully applied our framework on citrate synthases and hexokinases (Figure S6).

A major limitation of our framework is that finetuning ProtGPT2 requires a large GPU memory: among the NVIDIA GPU hardware we have tested (A100, A10, V100, T4, P4, P100), only NVIDIA A100 (80G) was able to handle the workload. A minor disadvantage is that, unlike GAN requires only positive set, a negative set is also needed to train our discriminator. However, obtaining a negative set from protein sequence databases such as UniProt[34] is straightforward.

In summary, we developed a framework that can be quickly tailored to various peptide or protein tasks without the need to re-learn the protein grammar. Compared to other de novo peptide/protein design methods,[21,73,90] our framework is time- and data-efficient, and is more convenient to use without the need to incorporating multiple filters or classifiers for screening purposes. Therefore, this framework may have the potential to significantly accelerate AI-based DNPD.

ABBREVIATIONS

generative pre-trained transformers (GPT), natural language processing (NLP), generative adversarial network (GAN), language model (LM), recurrent neural networks (RNN), antimicrobial peptide (AMP), malate dehydrogenase (MDH), enzyme commission (EC), minimum inhibitory concentration (MIC).

REFERENCES

1. Linsky T. W., Vergara R., Codina N., et al. (2020). De novo design of potent and resilient hACE2 decoys to neutralize SARS-CoV-2. *Science* **370**:1208–1214. DOI:10.1126/science.abe0075
2. Sesterhenn F., Yang C., Bonet J., et al. (2020). De novo protein design enables the precise induction of RSV-neutralizing antibodies. *Science* **368**:eaay5051. DOI:10.1126/science.aay5051
3. Pan X., and Kortemme T. (2021). Recent advances in de novo protein design: Principles, methods, and applications. *J. Biol. Chem.* **296**:100558. DOI:10.1016/j.jbc.2021.100558
4. Quijano-Rubio A., Yeh H.-W., Park J., et al. (2021). De novo design of modular and tunable protein biosensors. *Nature* **591**:482–487. DOI:10.1038/s41586-021-03258-z
5. Liu K., Zhang Y., Liu K., et al. (2022). De novo design of a transcription factor for a progesterone biosensor. *Biosens. Bioelectron.* **203**:113897. DOI:10.1016/j.bios.2021.113897
6. Jackson C., Anderson A., and Alexandrov K. (2022). The present and the future of protein biosensor engineering. *Curr. Opin. Struct. Biol.* **75**:102424. DOI:10.1016/j.sbi.2022.102424
7. Madhavan A., Arun K., Binod P., et al. (2021). Design of novel enzyme biocatalysts for industrial bioprocess: Harnessing the power of protein engineering, high throughput screening and synthetic biology. *Bioresour Technol.* **325**:124617. DOI:10.1016/j.biortech.2020.124617
8. Ennist N. M., Zhao Z., Stayrook S. E., et al. (2022). De novo protein design of photochemical reaction centers. *Nat. Commun.* **13**:4937. DOI:10.1038/s41467-022-32710-5
9. Currin A., Swainston N., Day P. J., et al. (2015). Synthetic biology for the directed evolution of protein biocatalysts: navigating sequence space intelligently. *Chem. Soc. Rev.* **44**:1172–1239. DOI:10.1039/c4cs00351a
10. Ferruz N. and Höcker B. (2022). Controllable protein design with language models. *Nat. Mach. Intell.* **4**:521–532. DOI:10.1038/s42256-022-00499-z
11. Ferruz N., Heinzinger M., Akdel M., et al. (2022). From sequence to function through structure: deep learning for protein design. *Comput. Struct. Biotechnol. J.* **21**:238–250. DOI:10.1016/j.csbj.2022.11.014
12. Madani A., Krause B., Greene E. R., et al. (2023). Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* **41**:1–8. DOI:10.1038/s41587-022-01618-2
13. Min B., Ross H., Sulem E., et al. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys* **56**:1–40. DOI:10.1145/3605943
14. Budzianowski P. and Vulić I. (2019). Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*. DOI:10.48550/arXiv.1907.05774
15. Alexandr N., Irina O., Tatyana K., et al. (2021). Fine-tuning GPT-3 for Russian text summarization. Silhavy, R., Silhavy, P., Prokopova, Z. (eds). Data Science and Intelligent Systems. (Springer, Cham), pp:748–757. DOI:10.1007/978-3-030-90321-3_61
16. Liao Y., Wang Y., Liu Q., et al. (2019). Gpt-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151*. DOI:10.48550/arXiv.1907.00151.
17. Strokach A. and Kim P. M. (2022). Deep generative modeling for protein design. *Curr. Opin. Struct. Biol.* **72**:226–236. DOI:10.1016/j.sbi.2021.11.008
18. Wang Y., Deng J., Sun A., et al. (2022). Perplexity from PLM Is Unreliable for Evaluating Text Quality. *arXiv preprint arXiv:2210.05892*. DOI:10.48550/arXiv.2210.05892.
19. Newman D., Noh Y., Talley E., et al. (2010). Evaluating topic models for digital libraries. Proceedings of the 10th annual joint conference on Digital libraries, pp:215-224. DOI:10.1145/1816123.1816156
20. Chang J., Gerrish S., Wang C., et al. (2009). Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, pp:288-296.
21. Repecka D., Jauniskis V., Karpus L., et al. (2021). Expanding functional protein sequence spaces using generative adversarial networks. *Nat. Mach. Intell.* **3**:324–333. DOI:10.1038/s42256-021-00310-5
22. Dauparas J., Anishchenko I., Bennett N., et al. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. *Science* **378**:49–56. DOI:10.1126/science.add2187
23. Yu T., Cui H., Li J. C., et al. (2023). Enzyme function prediction using contrastive learning. *Science* **379**:1358–1363. DOI:10.1126/science.adf2465
24. Nivón L. G., Moretti R., and Baker D. (2013). A Pareto-optimal refinement method for protein design scaffolds. *PloS one* **8**:e59004. DOI:10.1371/journal.pone.0059004
25. Meier J., Rao R., Verkuil R., et al. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems* **34**:29287–29303.
26. Yang K. K., Fusi N., and Lu A. X. (2022). Convolutions are competitive with transformers for protein sequence pretraining. *Cell Syst.* **15**:286–294.e2. DOI:10.1016/j.cels.2024.01.008
27. Yang K. K., Zanichelli N., and Yeh H. (2022). Masked inverse folding with sequence transfer for protein representation learning. *Protein Eng. Des. Sel.* **36**:gzad015. DOI:10.1093/protein/gzad015
28. Hsu C., Verkuil R., Liu J., et al. (2022). Learning inverse folding from millions of predicted structures. International Conference on Machine Learning. *Proceedings of the 39th International Conference on Machine Learning* **162**:8946-8970.
29. Johnson S. R., Fu X., Viknander S., et al. (2025). Computational scoring and experimental evaluation of enzymes generated by neural networks. *Nat. Biotechnol.* **43**:396–405. DOI:10.1038/s41587-024-02214-2
30. Villegas-Morcillo A., Makrodimitris S., van Ham R. C., et al. (2021). Unsupervised

protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics* **37**:162–170. DOI:10.1093/bioinformatics/btaa701
31. Heinzinger M., Elnaggar A., Wang Y., et al. (2019). Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics* **20**:1–17. DOI:10.1186/s12859-019-3220-8
32. Jha K., Saha S., and Singh H. (2022). Prediction of protein–protein interaction using graph neural networks. *Sci. Rep.* **12**:8360. DOI:10.1038/s41598-022-12201-9
33. Singh S., Chaudhary K., Dhanda S. K., et al. (2016). SATPdb: a database of structurally annotated therapeutic peptides. *Nucleic Acids Res.* **44**:D1119–D1126. DOI:10.1093/nar/gkv1114
34. Consortium T. U. (2021). UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**:D480–D489. DOI:10.1093/nar/gkaa1100
35. Sidorczuk K., Gagat P., Pietluch F., et al. (2022). Benchmarks in antimicrobial peptide prediction are biased due to the selection of negative data. *Briefings Bioinform.* **23**:bbac343. DOI:10.1093/bib/bbac343
36. McDonald A. G., and Tipton K. F. (2021). Enzyme nomenclature and classification: The state of the art. *FEBS J.* **290**:2214–2231. DOI:10.1111/febs.16274
37. Li W. and Godzik A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**:1658–1659. DOI:10.1093/bioinformatics/btl158
38. Wolf T., Debut L., Sanh V., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*. DOI:10.48550/arXiv.1910.03771
39. Elnaggar A., Heinzinger M., Dallago C., et al. (2020). ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*. DOI:10.48550/arXiv.2007.06225
40. Chollet F. (2015). Keras: Deep learning library for theano and tensorflow. https://keras.io
41. Agarap A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*. DOI:10.48550/arXiv.1803.08375
42. Rasamoelina A. D., Adjailia F., and Sinčák P. (2020). A review of activation function for artificial neural network. 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE. DOI:0.1109/SAMI48414.2020.9108717.
43. Kingma D. P., and Ba J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. DOI:10.48550/arXiv.1412.6980
44. Mistry J., Chuguransky S., Williams L., et al. (2021). Pfam: The protein families database in 2021. *Nucleic Acids Res.* **49**:D412–D419. DOI:10.1093/nar/gkaa913
45. Eddy S. R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.* **7**:e1002195. DOI:10.1371/journal.pcbi.1002195
46. Camacho C., Coulouris G., Avagyan V., et al. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics* **10**:1–9. DOI:10.1186/1471-2105-10-421
47. Mahlich Y., Steinegger M., Rost B., et al. (2018). HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics* **34**:i304–i312. DOI:10.1093/bioinformatics/bty262
48. Hinton G. E. and Roweis S. (2002). Stochastic neighbor embedding. *Advances in neural information processing systems*, pp:857 - 864.
49. Pedregosa F., Varoquaux G., Gramfort A., et al. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **12**:2825–2830.
50. Burley S. K., Bhikadiya C., Bi C., et al. (2023). RCSB Protein Data Bank (RCSB. *org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. Nucleic Acids Res.* **51**:D488–D508. DOI:10.1093/nar/gkac1077
51. Katoh K., Rozewicki J. and Yamada K. D. (2019). MAFFT online service: Multiple sequence alignment, interactive sequence choice and visualization. *Brief. Bioinform.* **20**:1160–1166. DOI:10.1093/bib/bbx108
52. Bodenhofer U., Bonatesta E., Horejš-Kainrath C., et al. (2015). msa: An R package for multiple sequence alignment. *Bioinformatics* **31**:3997–3999. DOI:10.1093/bioinformatics/btv494
53. Yuan S., Chan H. S., and Hu Z. (2017). Using PyMOL as a platform for computational drug design. *Wiley Interdiscip. Rev.:Comput. Mol. Sci.* **7**:e1298. DOI:10.1002/wcms.1298
54. Koh H. Y., Nguyen A. T., Pan S., et al. (2024). Physicochemical graph neural network for learning protein–ligand interaction fingerprints from sequence data. *Nat. Mach. Intell.* **6**:673–687. DOI:10.1038/s42256-024-00847-1
55. Koes D. R., Baumgartner M. P., and Camacho C. J. (2013). Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J. Chem. Inf. Model.* **53**:1893–1904. DOI:10.1021/ci300604z
56. Lin Z., Akin H., Rao R., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**:1123–1130. DOI:10.1126/science.ade2574
57. Robin X., Turck N., Hainard A., et al. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* **12**:1–8. DOI:10.1186/1471-2105-12-77
58. Xu J., Li F., Leier A., et al. (2021). Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides. *Brief. Bioinform.* **22**:bbab083. DOI:10.1093/bib/bbab083
59. Devlin J., Chang M.-W., Lee K., et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. DOI:10.48550/arXiv.1810.04805
60. Palermo E. F., and Kuroda K. (2010). Structural determinants of antimicrobial activity in polymers which mimic host defense peptides. *Appl. Microbiol. Biotechnol.* **87**:1605–1615. DOI:10.1007/s00253-010-2687-z
61. Shagaghi N., Palombo E. A., Clayton A. H., et al. (2018). Antimicrobial peptides: biochemical determinants of activity and biophysical techniques of elucidating their functionality. *World J. Microbiol. Biotechnol.* **34**:1–13. DOI:10.1007/s11274-018-2444-5
62. Kumar P., Kizhakkedathu J. N., and Straus S. K. (2018). Antimicrobial peptides: diversity, mechanism of action and strategies to improve the activity and biocompatibility in vivo. *Biomolecules* **8**:4. DOI:10.3390/biom8010004
63. Ma Y., Guo Z., Xia B., et al. (2022). Identification of antimicrobial peptides from the human gut microbiome using deep learning. *Nat. Biotechnol.* **40**:921–931. DOI:10.1038/s41587-022-01226-0
64. Nagarajan D., Nagarajan T., Roy N., et al. (2018). Computational antimicrobial peptide design and evaluation against multidrug-resistant clinical isolates of bacteria. *J. Biol. Chem.* **293**:3492–3509. DOI:10.1074/jbc.M117.805499
65. Rosano G. L., and Ceccarelli E. A. (2014). Recombinant protein expression in Escherichia coli: advances and challenges. *Front. Microbiol.* **5**:172. DOI:10.3389/fmicb.2014.00172
66. Roope L. S., Smith R. D., Pouwels K. B., et al. (2019). The challenge of antimicrobial resistance: what economics can contribute. *Science* **364**:eaau4679. DOI:10.1126/science.aau4679
67. Bahar A. A. and Ren D. (2013). Antimicrobial peptides. *Pharmaceuticals* **6**:1543–1575. DOI:10.3390/ph6121543
68. Chen X., Li C., Bernards M. T., et al. (2021). Sequence-based peptide identification, generation, and property prediction with deep learning: A review. *Mol. Syst. Des. Eng.* **6**:406–428. DOI:10.1039/D0ME00161A
69. Yan J., Cai J., Zhang B., et al. (2022). Recent progress in the discovery and design of antimicrobial peptides using traditional machine learning and deep learning. *Antibiotics* **11**:1451. DOI:10.3390/antibiotics11101451
70. Ramazi S., Mohammadi N., Allahverdi A., et al. (2022). A review on antimicrobial peptides databases and the computational tools. *Database* **2022**:baac011. DOI:10.1093/database/baac011
71. Capecchi A., Cai X., Personne H., et al. (2021). Machine learning designs non-hemolytic antimicrobial peptides. *Chem. Sci.* **12**:9221–9232. DOI:10.1039/D1SC01713F
72. Dean S. N., Alvarez J. A. E., Zabetakis D., et al. (2021). PepVAE: variational autoencoder framework for antimicrobial peptide generation and activity prediction. *Front. Microbiol.* **12**:725727. DOI:10.3389/fmicb.2021.725727
73. Das P., Sercu T., Wadhawan K., et al. (2021). Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nat. Biomed. Eng.* **5**:613–623. DOI:10.1038/s41551-021-00689-x
74. Tucs A., Tran D. P., Yumoto A., et al. (2020). Generating ampicillin-level antimicrobial peptides with activity-aware generative adversarial networks. *ACS Omega* **5**:22847–22851. DOI:10.1021/acsomega.0c02088
75. Jang W. D., Kim G. B., Kim Y., et al. (2022). Applications of artificial intelligence to enzyme and pathway design for metabolic engineering. *Curr. Opin. Biotechnol.* **73**:101–107. DOI:10.1016/j.copbio.2021.07.024
76. Pleiss J. (2011). Protein design in metabolic engineering and synthetic biology. *Curr. Opin. Biotechnol.* **22**:611–617. DOI:10.1016/j.copbio.2011.03.004
77. Khurana S., Rawi R., Kunji K., et al. (2018). DeepSol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics* **34**:2605–2613. DOI:10.1093/bioinformatics/bty166
78. Han X., Wang X., and Zhou K. (2019). Develop machine learning-based regression predictive models for engineering protein solubility. *Bioinformatics* **35**:4640–4646. DOI:10.1093/bioinformatics/btz294
79. Li F., Yuan L., Lu H., et al. (2022). Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. *Nat. Catal.* **5**:662–672. DOI:10.1038/s41929-022-00798-z
80. Yu H., and Luo X. (2022). Pretrained language models and weight redistribution achieve precise kcat prediction. *bioRxiv*:2022.2011. 2023.517595. DOI:10.1101/2022.11.23.517595.
81. Hu X., Feng C., Ling T., et al. (2022). Deep learning frameworks for protein-protein interaction prediction. *Comput. Struct. Biotechnol. J.* **20**:3223–3233. DOI:10.1016/j.csbj.2022.06.025
82. Casadio R., Martelli P. L., and Savojardo C. (2022). Machine learning solutions for predicting protein–protein interactions. *Wiley Interdiscip. Rev.:Comput. Mol. Sci.* **12**:e1618. DOI:10.1002/wcms.1618
83. Yeh A. H.-W., Norn C., Kipnis Y., et al. (2023). De novo design of luciferases using deep learning. *Nature* **614**:774–780. DOI:10.1038/s41586-023-05696-3
84. Watson J. L., Juergens D., Bennett N. R., et al. (2023). De novo design of protein structure and function with RFdiffusion. *Nature* **620**:1089–1100. DOI:10.1038/s41586-

023-05696-3
85. Yu L., Zhang W., Wang J., et al. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. Proceedings of the AAAI conference on artificial intelligence.DOI:10.1609/aaai.v31i1.10804
86. Arjovsky M., and Bottou L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*. DOI:10.48550/arXiv.1701.04862
87. Caccia M., Caccia L., Fedus W., et al. (2018). Language gans falling short. *arXiv preprint arXiv:1811.02549*. DOI:10.48550/arXiv.1811.02549
88. Thanh-Tung H., and Tran T. (2020). Catastrophic forgetting and mode collapse in gans. 2020 international joint conference on neural networks (ijcnn). IEEE. DOI:10.1109/IJCNN48605.2020.9207181
89. Shmelkov K., Schmid C., and Alahari K. (2018). How good is my GAN? *Proceedings of the European conference on computer vision (ECCV)*, pp:218 - 234. DOI:10.1007/978-3-030-01216-8_14
90. Szymczak P., Możejko M., Grzegorzek T., et al. (2023). Discovering highly potent antimicrobial peptides with deep generative model HydrAMP. *Nat. Commun.* **14**:1453. DOI:10.1038/s41467-023-36994-z

FUNDING AND ACKNOWLEDGMENTS

We would like to thank Yu Han (SIAT) for the constructive discussion and all researchers who made the data and tools relevant to this study available. This project is supported by National Key R&D Program of China (2019YFA0904100), National Natural Science Foundation of China (32071421), Guangdong Basic and Applied Basic Research Foundation (2021B1515020049 and 2022A1515110639), Shenzhen Science and Technology Program (KJZD20230923115906013, JCYJ20220531100207017 and RCYX 20200714114736026), SIAT Distinguished Young Scholars (E4G021). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

AUTHOR CONTRIBUTIONS

Z.Z. and X.L. conceived the ideas; Z.Z. performed the modeling, data analysis, and result visualization; R.X. performed the AMP and MDH experimental validation and wrote the methods for it; Z.Z. wrote all other parts of the manuscript; J.G. performed preliminary finetuning experiments and additional case demonstration; J.J. conducted the protein-ligand affinity prediction; X.L. and H.H. revised the manuscript.

DECLARATION OF INTERESTS

X.L. has a financial interest in Demetrix and Synceres. X.L. is an Editorial Board member of The Innovation Life. He was blinded from reviewing or making final decisions on the manuscript. The article was subject to the journal's standard procedures, with peer review handled independently of this Editorial Board member and the research groups. The other authors declare that they have no conflicts of interest.

ETHICAL STATEMENT AND PATIENT CONSENT

Not applicable.

DATA AND CODE AVAILABILITY

Codes for the framework and result analysis, as well as the data to replicate this work, were deposited in GitHub repository (https://github.com/zishuozeng/GPT_protein_design).

SUPPLEMENTAL INFORMATION

It can be found online at https://doi.org/10.59717/j.xinn-life.2025.100133.