ACS Catalysis

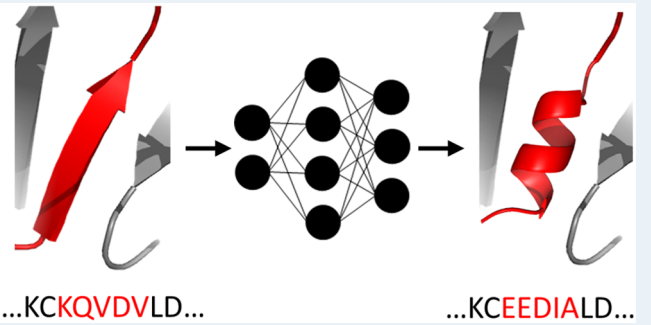
pubs.acs.org/acscatalysis
Perspective

Machine Learning-Guided Protein Engineering

Petr Kouba,[#] Pavel Kohout,[#] Faraneh Haddadi,[#] Anton Bushuiev, Raman Samusevich, Jiri Sedlar, Jiri Damborsky, Tomas Pluskal,* Josef Sivic,* and Stanislav Mazurenko*

Cite This: ACS Catal. 2023, 13, 13863−13895

ABSTRACT: Recent progress in engineering highly promising biocatalysts has increasingly involved machine learning methods. These methods leverage existing experimental and simulation data to aid in the discovery and annotation of promising enzymes, as well as in suggesting beneficial mutations for improving known targets. The field of machine learning for protein engineering is gathering steam, driven by recent success stories and notable progress in other areas. It already encompasses ambitious tasks such as understanding and predicting protein structure and function, catalytic efficiency, enantioselectivity, protein dynamics, stability, solubility, aggregation, and more. Nonetheless, the field is still evolving, with many challenges to overcome and questions to address. In this Perspective, we provide an overview of ongoing trends in this domain, highlight recent case studies, and examine the current limitations of machine learning-based methods. We emphasize the crucial importance of thorough experimental validation of emerging models before their use for rational protein design. We present our opinions on the fundamental problems and outline the potential directions for future research.



KEYWORDS: *activity, artificial intelligence, biocatalysis, deep learning, protein design*

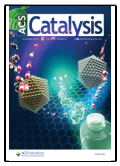
1. INTRODUCTION

Biocatalysis is a promising field that offers diverse possibilities for creating sustainable and environmentally friendly solutions in various industries. Its potential stems from its ability to mimic and harness the power of nature using cells and enzymes that have evolved over millions of years to perform specific chemical reactions with high efficiency. This makes it possible to transform chemical compounds selectively and efficiently, providing an alternative to traditional chemical catalysis, which often requires harsh conditions and toxic chemicals.[1] Biocatalysts could therefore be valuable in the production of fine chemicals, pharmaceuticals, and food ingredients as well as in the development of sustainable processes for the production of energy and materials.[2] In addition, biocatalysis is an exciting area of research and development with great promise for the future because of the potential to unlock new solutions for diverse challenges by providing green alternatives to traditional chemical processes, new energy sources, and tools for improving the overall efficiency of industrial processes or biological removal of recalcitrant waste.[3−5] It is also a highly interdisciplinary research area that makes heavy use of advanced experimental techniques and computational methods.[6]

Many research fields are undergoing a gradual transition from near-exclusive reliance on experimental work to hybrid approaches that incorporate computational simulations and data-driven methods.[7−9] In the past, researchers would accumulate observations from individual experiments and use the resulting data to formulate fundamental rules. They then created simulations based on these rules to better understand the system under investigation. As computational power has increased, researchers have been able to shift toward data-driven methods that rely on **machine learning** (ML, see the glossary in Table 1 for terms in bold) algorithms to deduce rules directly from data.[10,11] This transition has made it possible to efficiently and comprehensively analyze large and complex data sets that are often generated by high-throughput technologies. Particularly, very powerful **deep learning** algorithms are finding a wide range of applications in life sciences and will be discussed in great detail in this Perspective. While experimental science and computational simulations still play essential roles, the trend toward data-driven methods will likely continue as technology and data collection methods evolve further.[7]

This paradigm shift is illustrated by the exponentially growing number of scientific articles describing the use of

Received: June 15, 2023
Revised: September 20, 2023
Published: October 13, 2023



https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

Table 1. Glossary of Terms Used Frequently in the Context of Machine Learning for Enzyme Engineering[a]

accuracy	metric primarily used for classification tasks, measuring the ratio of correct predictions to all predictions produced by a machine learning model
active learning	a type of machine learning in which the learning algorithm queries a person for providing labels for particular data points during training iteratively. the first iteration usually starts with many unlabeled and few labeled data points, *e.g.*, protein sequences. after training on this data set, the algorithm proposes a next set of data points for labeling to the experimenter, *e.g.*, more sequences to be characterized in the lab. their labels are then provided to the algorithm for the next iteration, and the cycle repeats several times
artificial intelligence (AI)	artificial intelligence as defined by McCarthy is "the science and engineering of making intelligent machines, especially intelligent computer programs"
continual learning	a concept in which a model can train on new data while maintaining the abilities acquired from earlier training on old data
cross-validation (*K*-fold)	an approach to evaluating the performance of a model whereby a data set is split into *K* parts, the model is retrained *K* times on all but one part, and the performance is evaluated on the excluded part. this way each data point is used once for validation, and *K* different evaluations are produced to provide a distribution of the values
deep learning (DL)	a branch of machine learning that uses multiple-layer neural network architectures. deep networks generally include many more parameters (sometimes, billions) and hyperparameters than traditional machine learning models. this gives deep neural networks tremendous expressive power and design flexibility, which has made them a major driver of modern technology with applications ranging from on-the-fly text generation to protein structure prediction
diffusion	deep learning paradigm based on denoising diffusion probabilistic modeling. diffusion models learn to generate novel objects, *e.g.*, images or proteins, by reconstructing artificially corrupted training examples
embedding	representation of high-dimensional data, *e.g.*, text, images, or proteins, in a lower-dimensional vector space while preserving important information
end-to-end learning	a type of ML that requires minimal to no data transformation (*e.g.*, just one-hot encoding of the input) to train a predictor. This is often the case in deep learning, when abundant data are available to establish direct input-to-output correspondence, in contrast to classical ML approaches using small data sets, which typically require feature and label engineering before the data can be used for training
equivariance	an ML model is said to be equivariant with respect to a particular transformation if the order of applying the transformation and the model to an input does not change the outcome. for example, if we pass a rotated input to a model that is equivariant to rotation, the result will be the same as if the model was applied to the original input and the output was then rotated
explainable artificial intelligence (XAI)	AI or ML-based algorithms designed such that humans can understand the reasons for their predictions. its core principles are transparency, interpretability, and explainability
findable, accessible, interoperable, reusable (FAIR principles)	principles for the management and stewardship of scientific data to ensure findability, accessibility, interoperability, and reusability
fine-tuning	an approach to transfer learning (see "transfer learning" below) in which all or part of the weights of an artificial neural network pretrained on another task are further adjusted ("fine-tuned") during the training on a new task
generalizability	in the context of ML models, generalizability refers to a model's ability to perform well on new data not used during the training process
generative models	algorithms aiming to capture the data distribution of the training samples to be capable of generating novel samples resembling the data that they was trained on
inductive bias	set of assumptions of a model used for predictions over unknown inputs. for example, the model can be built such that it can only predict values within a certain range consistent with the expected range of values for a particular problem
learning rate	a parameter influencing the speed of the training of an ML model. a higher learning rate increases the effect of a single pass of the training data on the model's parameters
loss function/cost function	a function used to evaluate a model during training. by iteratively minimizing this function, the model updates the values of its parameters. a typical example is mean-squared error of prediction
machine learning (ML)	machine learning, according to Mitchell, is the science that is "concerned with the question of how to construct computer programs that automatically improve with experience". the terms ML and AI are often used interchangeably, but such usage is an oversimplification, as ML involves learning from data whereas AI can be more general
masking	deep learning paradigm for self-supervised learning. neural networks trained in a masked-modeling regime acquire powerful understanding of data by learning to reconstruct masked parts of inputs, *e.g.*, masked words in a sentence, numerical features artificially set to zeros, or hidden side-chains in a protein structure
multilayer perceptron (MLP)	a basic architecture of artificial neural networks. it consists of multiple fully connected layers of neurons. each neuron calculates a weighted sum of inputs from the previous layer, applies a nonlinear activation function, and sends the result to the neurons in the next layer
multiple sequence alignment (MSA)	a collection of protein or nucleic acid sequences aligned based on specific criteria, *e.g.*, allowing introductions of gaps with a given penalty or providing substitution values for pairs of residue types, to maximize similarity at aligned sequence positions. sequence alignments provide useful insights into the evolutionary conservation of sequences
one-hot encoding for protein sequence	each amino acid residue is represented by a 20-dimensional vector with a value of one at the position of the corresponding amino acid in the 20-letter alphabet and zeros elsewhere
overfitting	case of an inappropriate training of ML model in which the model has too many degrees of freedom, and it is allowed to use these degrees of freedom to fit the noise in the training data during training. as a result, the model can reach seemingly excellent performance on the training data, but it will fail to generalize to new data
regularization	in ML, regularization is a process used to prevent model overfitting. regularization techniques typically include adding a penalty on the magnitude of model parameters into the loss function to favor the use of low parameter magnitudes and thereby compress the parameter space. another popular regularization method for DL is to use a dropout layer, which randomly switches network neurons on and off during training
reinforcement learning (RL)	a machine learning paradigm in which the problem is defined as finding an optimal sequence of actions in an environment where the value of each action is quantified using a system of rewards. for example, if RL were used to learn to play a boardgame, the model would function as a player, the actions would represent the moves available to the player, and the game would constitute the environment. using game simulations, the model learns to perform stronger actions based on the "rewards" (feedback from the environment, *e.g.*, winning/losing the game) it receives for its past actions

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

machine learning for protein engineering (Figure 1). The trend toward data-driven methods is expected to continue as

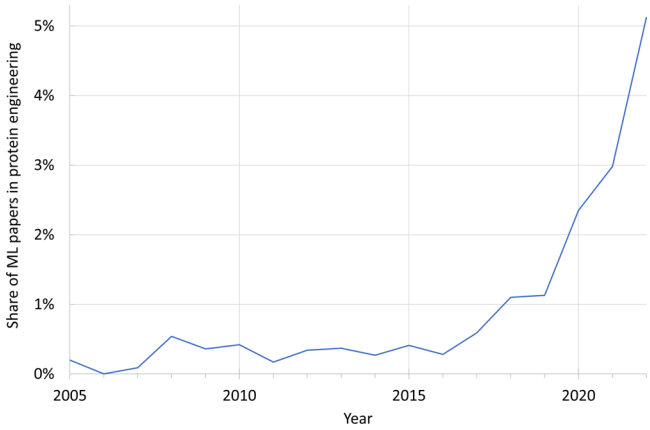


Figure 1. The trend in the use of machine learning in the literature on protein engineering. The graph shows the ratio of publications mentioning "machine learning" and "protein engineering" to all papers mentioning "protein engineering" in their title, abstract, or keywords, based on the Scopus database. This trend illustrates the increasing attention ML is receiving as a generally applicable and useful technology for protein engineering.

technological advances allow us to accumulate, deposit, and reuse biological and biochemical data more effectively. This is being facilitated by initiatives such as the **FAIR principles**, which promote the findability, accessibility, interoperability, and reusability of data, and the European Open Science Cloud, which is designed to promote best practices in handling data.[12] These large-scale initiatives are expected to accelerate the adoption of data-driven methods by making it easier for researchers to access, use, and share existing data and ensuring that these data are of high quality.

This Perspective focuses on the application of machine learning to protein engineering, which means improving the properties of biocatalysts by optimizing their sequences and tertiary structures using molecular biology techniques. Time-wise, we will primarily cover the period since our previous review on the same topic, published in 2019.[13] As for particular areas, we will be focusing mainly on the applications regarding engineering by mutating known proteins rather than designing proteins *de novo*. While readers with a particular interest in *de novo* design might also find this Perspective helpful, as we cover many techniques common for various protein design tasks, for particular details on *de novo* design, such as deep **generative modeling**, we refer to other reviews.[14−16] We will also introduce high-level concepts from machine learning to familiarize the reader with a broader context and will not cover in depth technical aspects such as specifics of various neural network architectures. We refer the readers to several excellent recent reviews on those topics.[10,17−23] We will consider new methods from a user's perspective. We find this important because the methods presented in research papers, while being exciting and innovative, often have only a limited impact if the wider community cannot quickly and easily adopt them. Moreover, we will draw inspiration from other domains, as we believe that identifying parallels between tasks in different fields can accelerate the development of more powerful and practically useful methods. In particular, by examining how solutions have been developed in other disciplines, we can

Table 1. continued

self-supervised learning	a machine learning paradigm of utilization of unlabeled data in a setting of supervised learning. In self-supervised learning, the key is to define a proxy task for which labels can be synthetically generated for the unlabeled data, and then the task can be learned in a supervised manner. for example, in natural language processing, a popular task is to take sentences written in natural language, mask (see "masking" above) words in those sentences, and learn to predict the missing word. such a proxy task can help the model to learn the distribution of the data it is supposed to work with
semisupervised learning	a machine learning paradigm for tasks where the amount of labeled data is limited but there is an abundance of unlabeled data available. the unlabeled data are used to learn a general distribution of the data, aiding the learning of a supervised model. for example, all the data can be clustered by an unsupervised algorithm, and the unlabeled samples can be automatically labeled based on the labels present in the cluster, leading to enhancement of the data set for supervised learning, which can benefit its performance despite the lower quality of labeling
supervised learning	a machine learning paradigm in which the goal is to predict a particular property known as a label for each data point. for example, the datapoints can be protein sequences and the property to be predicted would be the solubility (soluble/insoluble). training a model in a supervised way requires having the training data equipped with labels
transfer learning	a machine learning technique in which a model is first trained for a particular task and then used ("transferred") as a starting point for a different task. some of the learned weights can further be tuned to the new task (see "fine-tuning" above), or the transferred model can be used as a part of a new model that includes, for example, additional layers trained for the new task
transformer	transformers learn to perform complex tasks by deducing how all parts of input objects, *e.g.*, words in a sentence or amino acids in a protein sequence, are related to each other, using a mechanism called "attention". the transformer architecture is currently one of the most prominent neural network architectures
underfitting	the case of an insufficient training of a ML model, where the model could not capture the patterns in the available data well and exhibits high training error. it can be caused for example by a wrongly chosen model class, too strong regularization, an inappropriate learning rate, or too short training time
unsupervised learning	a machine learning paradigm in which the goal is to identify patterns in unlabeled data and the data distribution. Typical examples of unsupervised learning techniques include clustering algorithms and data compression or projection methods such as principal component analysis. the advantage of these methods is the capability of handling unlabeled data, often at the expense of their predictive power

[a]Focusing on their meaning in the context of this Perspective. The terms from this table are highlighted in bold upon their first usage in the text.

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

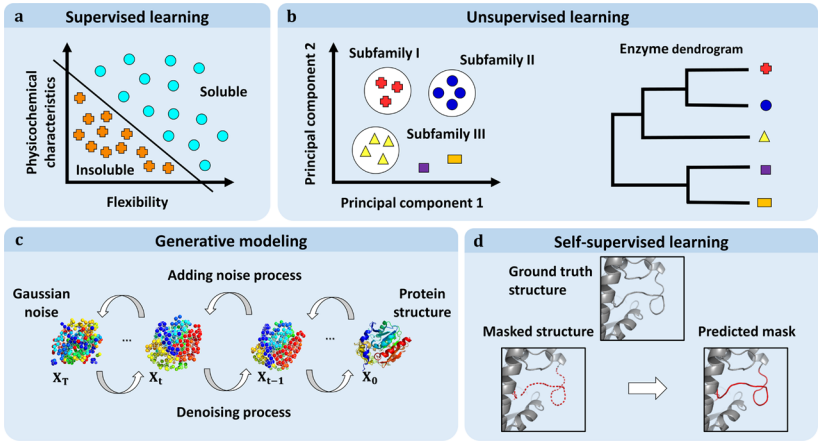


Figure 2. Four main categories of machine learning methods. (a) **Supervised learning** methods use labels. For example, each protein in a data set might be labeled "soluble" or "insoluble", and the model would then aim to find the optimal decision boundary between these two classes in the feature space. The learned boundary is then used to make predictions about new data for which labels are unavailable. (b) **Unsupervised learning** methods typically find patterns, *e.g.*, clusters or groups) in unlabeled data. Examples include clustering enzymes into subfamilies or grouping them into a dendrogram. (c) **Generative models** learn the distribution of the training data to generate new instances corresponding to that distribution. Models of this type include diffusion models, which are trained to denoise synthetically noised inputs. The trained models can then be applied to random noise on the input to create a sample resembling the training data, *e.g.*, a new protein structure. (d) **Self-supervised learning** methods transform an unsupervised problem into a supervised problem, for example, by masking a part of a sequence or structure (red dashed loop) and then predicting the masked information (red loop). Such proxy tasks enable the model to learn important characteristics of the underlying data and can lead the model to perform well even on different tasks.

learn effective ways of making methods and software tools applicable to a broader range of users.[24,25]

This paper is organized as follows. Section 2 briefly reviews the basics of machine learning and underlines the similarities and differences between data related to proteins and data in other domains. Section 3 provides a comprehensive review of machine learning in protein engineering and highlights recent progress in the field. In section 4, we examine a series of exciting recent case studies in which machine learning methods were applied to create new enzyme designs for use in the laboratory and in practical applications. In section 5, we identify gaps in the field that remain to be filled. Finally, in section 6, we investigate what inspiration we can draw from other disciplines to bring ML-based enzyme design to a new level.

2. PRINCIPLES OF MACHINE LEARNING

As some of our readers might be unfamiliar with machine learning (ML), we start with a brief introduction to the topic. We will cover the basics of ML-based pipelines and vocabulary, highlight similarities between protein engineering and other domains from a ML perspective, outline the main traits that distinguish protein data from other data types regularly used in ML, and summarize the challenges of performing ML with protein data.

2.1. Machine Learning Basics. Machine learning is often seen as a subcategory of **artificial intelligence** (AI). Its primary purpose is to learn patterns directly from available data and use the learned patterns to generate predictions for new data. Its main difference from other methods for modeling a system's behavior, such as quantum mechanical calculations, is that ML does not rely on hard-coded rules to make predictions. Instead, ML models are mathematical functions that depend on generic parameters whose values are obtained (learned) through optimization using available data and an optimization criterion, the so-called **loss function**.

Since the final model is derived from the input data, careful data collection is vitally important for machine learning. In particular, any biases, measurement noise, and imbalances must be recognized and accounted for. Moreover, as ML is based on mathematical functions, every data point in a data set typically needs to be represented as a vector of numbers that are commonly referred to as **features**. Features may be obtained by simple encoding of the raw data, *e.g.*, **end-to-end learning** and **one-hot encoding**, but they may also represent more involved quantities derived from the raw data. For example, when predicting solubility from a protein sequence, the features may be simple amino acid counts, propensities of different residues to form secondary structures, conservation scores of proteins, or variables representing aggregated physicochemical properties.[26] Choosing informative and discriminating features that provide relevant information about the underlying pattern in the data is crucial in ML because the features are the only data characteristics that the algorithm will exploit during training and when making predictions based on future inputs.

Several different categories of ML problems exist. In **supervised learning** problems, the goal is to predict a particular property (known as a label) for each data point (Figure 2A). For example, if we were seeking to predict protein solubility, each data point could be labeled "soluble" or "insoluble" based on experimental results. Data points can have multiple labels, so a given protein could have the labels "soluble", "from a thermophilic organism", and "globular". Labels can form a set of classes or fall within a range of numerical values, giving rise to two subtypes of supervised learning problems: classification problems involving labels with no inherent order (*e.g.*, "soluble" or "insoluble") and regression problems involving labels corresponding to numerical values (*e.g.*, protein yields). In contrast, the goal in **unsupervised learning** problems is to identify patterns in unlabeled data. Unsupervised learning techniques include clustering algorithms and data compression or projection methods, such as principal

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

component analysis (Figure 2B). **Semisupervised learning** problems are those where the amount of labeled data is limited but there is an abundance of unlabeled data available. The unlabeled data are used to learn a general distribution of the data, aiding the learning of a supervised model. For example, all the data can be clustered by an unsupervised algorithm, and the unlabeled samples can be automatically labeled based on the labels present in the cluster, leading to the enhancement of the data set for supervised learning, which can benefit its performance despite the lower quality of labeling.

The boundary between supervised and unsupervised machine learning has been blurred by the emergence of methods that can create labels synthetically. For example, in a data compression method, the label might be the input itself and the algorithm may impose constraints (*e.g.*, a bottleneck in the architecture) that force the model to learn a more compact way of representing the data and their distribution. The algorithms that aim to capture the data distribution to generate new samples belong to a class of ML models called generative models. The most recent examples of this class include **diffusion models**, which have recently been used to generate protein backbone structures[27,28] and predict the binding of a flexible ligand to a protein[29] (Figure 2C). In diffusion models, synthetic training data are generated by gradually noising true data (X_0) in a stepwise manner to ultimately obtain a maximally noised sample (X_T). The sequences of increasingly noisy data are then reversed and used to train a model to denoise each individual step by having the (less noisy) sample X_{t-1} serve as a "label" for the (more noisy) following sample X_t. For more details on diffusion generative models and their applications in bioinformatics, see the recent review.[30] Alternatively, we can avoid the need for labels by **masking** a part of the input, *e.g.*, a residue in a protein sequence or structure,[31−33] and training a model that will predict the masked part. In other words, the original data (*e.g.*, the amino acid that was masked) are treated as the label for the corresponding data point. Such approaches belong to the methods of **self-supervised learning** (Figure 2D) and are currently attracting considerable attention because of their great success in large language models; it turns out that this "self-supervision" approach allows algorithms to learn useful characteristics of the data, such as grammar and semantics in the case of natural language models. The following sections present some applications of self-supervised learning in the enzymology domain.

In supervised learning, **underfitting** and **overfitting** are two critical concepts that must always be considered. Underfitting refers to a situation in which either the selected class of models is insufficient to approximate patterns in the available data, the **regularization** is too strong, or the parameters of the training process such as the duration of the training or the **learning rate** were inappropriate. As a result, the model fails to capture the relationship between the input and output and has a high training error. Conversely, overfitting happens when a model has too many degrees of freedom, allowing it to start fitting noise in the training data during the training process. This leads to poor **generalization** and a significant drop in performance when the model is applied to new inputs. Robust evaluation of trained models is therefore crucial in machine learning to obtain feedback on the training process and develop improved training protocols or model hyperparameters.

The best practice in machine learning is to split the available data into three disjoint subsets: a training set, a validation set, and a test set. A model learns the underlying patterns within the data by fitting its parameters to the training set. The validation set is provided to the model at certain stages of the training for basic evaluation, and the results of these evaluations are used to select the model's hyperparameters. Finally, the test set is used to get a realistic estimate of the model's performance and is therefore only used once training has been completed and the final values for the hyperparameters have been set. Since the model does not see the test set during the training process, the model's performance, when applied to this "new" data set, should be similar to that achieved with the test set if the test set accurately represents the general distribution of the studied data.

The choice of evaluation metric depends on the task at hand. Classification problems are mainly evaluated based on **model accuracy**, *i.e.*, the ratio of correct predictions to the number of total predictions. Regression problems are typically evaluated based on the difference between the predicted and ground truth labels, so popular metrics include measures of the correlation between these labels as well as the mean squared error (MSE) and related metrics such as the root-mean-square error (RMSE). More complex problems often require customized metrics. For instance, in a protein structure prediction task, one might use the MSE between the predicted and actual locations of Cα atoms, which can be expressed using a fixed coordinate system (global alignment) or in terms of the local coordinates of each residue (local alignment). Both metrics indicate how well a predicted structure aligns with the corresponding actual structure.

2.2. Parallels between Machine Learning Tasks in Biocatalysis and Those in Other Domains.

One of the strengths of machine learning is its universality, as the algorithms used for protein engineering tasks are similar to those used in other domains. Therefore, scientists working with protein data can reuse and build upon existing solutions from other fields, such as natural language processing, computer vision, and network analysis.

Natural language processing (NLP) is a field of computer science that aims to teach computers how to understand and handle natural languages. New ML techniques have enabled great advances in this field in recent years. For instance, a common task in NLP is to generate semantically and grammatically correct sentences. In protein engineering, the strings of amino acids representing primary sequences can be regarded as words constructed from an alphabet of twenty letters representing the canonical amino acids. These words can represent secondary structures or other motifs that can be combined in meaningful ways to create sentences in the language of protein structure that correspond to functional proteins. Another common task in NLP is the assignment of labels to individual words (*e.g.*, to predict lexical categories or identify relevant information) or phrases (*e.g.*, for sentiment analysis). This data structure resembles that of annotated protein data sets with labels representing protein stability, binding affinity, specificity, or other characteristics.[34] Moreover, the complexities of the relationships between protein sequence, structure, and function are reminiscent of those in human languages, prompting researchers to adapt **transformer**-based large language models used in NLP to protein engineering tasks.[35]

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

Another field of computer science that has benefited greatly from recent advances in ML is computer vision, and techniques developed for use in this field have also found applications in the study of the protein structure. For example, protein structures can be converted to arrays of voxels (3D pixels) via the application of a discrete grid. The resulting representations are similar to those of volumetric 3D images, enabling the application of ML architectures, such as convolutional neural networks, that were originally designed to process image data. These networks learn representations through convolution and hierarchical aggregation and have recently been used to predict protein mutation landscapes,[31] protein−ligand binding affinity,[36] and the interactions of proteins with water molecules.[37] Denoising diffusion probabilistic models are another class of computer vision models that have been applied in protein structure prediction. They are trained to denoise existing samples, which allows them to generate novel samples by morphing random noise. This has led to major breakthroughs in image generation and the emergence of the highly successful models, such as DALL-E 2[38] or Stable Diffusion.[39] In protein science, such models have been used to perform fast protein−ligand binding,[29] generate new small-molecule ligands[40] and linkers,[41] and perform *de novo* design of large proteins.[27,28]

The parallels between images and protein structures can be further exploited by adapting techniques developed for video analysis to predict protein dynamics. For example, a trajectory generated in a molecular dynamics simulation can be regarded as a temporal sequence of 3D images. This makes protein dynamics analysis similar to video processing and implies that video methods for event detection[42] can be applied to molecular dynamics trajectories to detect events such as the opening of a tunnel.[43] Video processing techniques can thus be adapted to clarify a protein's function by analyzing the movement of individual atoms or groups of atoms within a protein structure in a manner similar to the movement of objects or groups of objects in a video. Moreover, there have been remarkable advances in ML techniques for video synthesis[44−46] that could inspire new methods for capturing and synthesizing protein dynamics.

Finally, one more domain is relevant to protein engineering: network analysis, which involves studying the properties and structures of interconnected elements. Network analysis techniques have been used successfully to study diverse social and biological networks, including the development of Covid-19-related sentiment on social networks[47] and protein−protein interaction networks.[48] Interactions between proteins or between a protein and a ligand can be represented as networks (graphs) in which nodes correspond to proteins and ligands while edges correspond to biological relationships between them. Once such a network has been defined, link prediction or community detection methods can be applied.[49] Alternatively, a protein structure can be represented as a network where nodes correspond to residues or individual atoms and edges correspond to inter-residue interactions or interatomic bonds. This makes it possible to use graph-based ML algorithms for tasks such as predicting protein function, solubility, or toxicity.[50,51] Moreover, data on protein interactomes and the structures of small molecules have been used to drive theoretical research on graph-based ML: the best-established benchmark for graph learning, OGB, includes multiple biochemical data sets of this type.[52]

2.3. Challenges of Machine Learning for Protein Data. As discussed above, there are striking similarities between protein engineering tasks and other ML domains, including natural language processing, computer vision, and network analysis. However, protein data also present unique challenges related to the representation of proteins, the construction of labeled data sets, and the establishment of robust training protocols.

The choice of protein representation is a key step in all protein-related computational tasks. Proteins can be represented at different levels of detail, from a discrete and accurate 1D representation of their amino acid sequence to a continuous and less accurate 3D representation of every atom position, including or excluding chemical bonds. The selected representation determines the type and amount of information available to the computational model and the range of applicable model architectures.

The go-to representation of a protein sequence is a string (word) constructed using an alphabet of 20 amino acids (letters). The length of the string equals the number of residues, and the n^{th} character encodes the amino acid at the n^{th} position in the protein sequence. *In silico*, the amino acids are typically represented using one-hot encoding. When the amount of data available for training is not enough for end-to-end learning, one-hot encoding of sequences can further be transformed into values corresponding to specific physicochemical characteristics of amino acids, *e.g.*, using AA indices.[53] These indices provide additional information and interpretability to the pipeline, although they were shown to perform on par with random vectors for some tasks.[54,55] Another strategy to enrich the protein sequence representation is to include evolutionary information using a **multiple sequence alignment** (MSA) instead of a single protein sequence. This evolutionary information is valuable in various tasks, most notably in structure prediction because the covariance of different residue positions in the sequence can be related to the residues' spatial proximity.[56]

The options for representing a protein's structure are more varied; representations may include all atoms, only selected chemical elements (*e.g.*, all atoms except hydrogens), or just key components of residues (*e.g.*, the α-carbons). Moreover, they may include different types of information about these atoms and/or residues. Ideally, to enable data-efficient model training, structural representations should be invariant to rotation, translation, and reflection. However, straightforward representations based on the 3D coordinates of the residues (atoms) lack this property. It is therefore common to represent protein structure using an inter-residue or interatomic distance matrix in which each row and column is assigned to a specific residue (or atom) of the protein and the value of each matrix entry is equal to the distance between the corresponding residues (atoms). Such a matrix is necessarily symmetric; therefore, it is common to take the upper (or lower) triangular part and convert it into a 1D vector for processing, *e.g.*, by a neural network. While this representation is rotationally and translationally invariant, it is also inherently redundant, as its spatial complexity is quadratic with respect to the number of residues (atoms).

Graph-based protein representations have recently attracted considerable attention.[57] A graph consists of a set of nodes linked by a set of edges. The nodes typically represent residues, atoms, or groups of spatially close atoms, while the edges usually correspond to chemical bonds, spatial proximity

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

(contacts) between the nodes, or both.[58] Graph-based protein representations are very flexible because the definition of the nodes and edges can be tailored to specific tasks and they can be made **equivariant** to rotation, translation, and reflection.[59] A convenient definition of the nodes and edges can also introduce an **inductive bias** that improves the model performance. For example, edges corresponding to chemical bonds can guide a model toward learning chemical knowledge more rapidly or with less data than would otherwise be needed. Graph neural network (GNN) architectures have recently achieved state-of-the-art performance in multiple protein-related tasks, as exemplified by the DeepFRI[60] and HIGH-PPI[61] methods for predicting protein function and protein−protein interactions, respectively. Special graph-based protein representations, such as point clouds (no edges) or complete graphs (a full set of edges), are especially convenient for processing using powerful transformer models.[62,63]

A more general approach to protein representation is to directly learn the representation by a deep learning model, a direction that is currently on the rise in biology.[64] Its aim is to remove the suboptimality of human-made choices by inferring the representation parameters from existing data. Furthermore, it is often possible to learn such a representation through self-supervised training, *i.e.*, without the need for annotated data. These representations can be obtained from the sequence data, *e.g.*, as done by the ESM ("Evolutionary Scale Modeling") language models,[65−67] as well as from large structural data sets, *e.g.*, as done by GearNet.[68] More and more models combine both sources of data, *e.g.*, ESM-GearNet.[32]

Obtaining appropriately labeled data sets can be challenging when seeking to apply ML in enzyme engineering because there is often a trade-off between data quality and quantity when selecting methods for acquiring experimental data. Most reliable biochemical methods using purpose-built instruments can only provide data for small numbers of protein variants[69] and are thus generally insufficient for representative sampling of vast mutational spaces. Conversely, high-throughput methods such as Deep Mutational Scanning (DMS) are prone to data quality issues[69] and face a throughput bottleneck in the case of enzymes whose screening is considerably slower than sequencing.[70]

The process of compiling new data sets from multiple sources can be complicated by the inconsistency of conventions in protein research. These inconsistencies include the differing biases of various experimental tools, the use of different distributions for data normalization, and inconsistent definitions of quantities such as stability, solubility, and enzyme activity. All of these can introduce errors into constructed data sets, for example, by causing contradictory labels to be applied to the same protein. Protein data may also contain biases introduced by the design strategy. For example, alanine tends to be overrepresented in mutational data due to the widely used alanine scanning technique.[71] It is important to consider these biases during data set construction and when interpreting model outputs since the composition of the training set significantly affects the space of patterns explored by the model.

A significant amount of protein data has been gathered over the years. However, much of these data are proprietary and thus inaccessible to the academic community. In addition, publicly available data sets are often published in an unstructured way, which limits their usability. While large language models such as GPT-3,[72] GPT-4,[73] or BioGPT[74] are remarkably effective at summarizing text on large scales, mining relevant data from publications still requires considerable human effort.

Some ML packages, *e.g.*, TorchProtein,[75] offer preprocessed data sets for various protein science tasks, making protein research more accessible to ML experts from other domains. Other packages, such as PyPEF[76] provide frameworks for the integration of simpler ML models together with special encodings derived from the AAindex database of physicochemical and biochemical properties of amino acids.[53] Despite this progress, the development of ML models for protein data requires a certain level of biochemistry domain knowledge to account for the specifics of protein data absent in other application areas of ML. These specifics include the evolutionary relationships and structural similarities between proteins. Models produced without the benefit of such expertise may lack practical utility due to data handling errors.

One common type of data handling error is data leakage between data splits. The training, validation, and test sets should not share the same (or nearly the same) data points because such overlaps could lead to over estimating the model's performance on new data, compromising the model's evaluation. In some data sets, all of the data points are distinct enough that randomly splitting the available data into disjoint sets is a viable strategy. However, more complex splitting strategies are often needed when dealing with protein data to avoid problems such as evolutionary data leakage.[77] It may also be important to consider multiple levels of separation when dealing with protein data, *e.g.*, consisting of mutations and their effects. For example, one might want to ensure that the same substitutions, positions, or proteins do not appear in the training and test sets. Defining protein similarity is also a challenging task for which multiple strategies exist. Many of these strategies involve clustering proteins based on sequence identity or similarity thresholds and then ensuring that all members of a given cluster are assigned to the same set when splitting the data. This strategy is particularly useful for constructing labeled data sets of protein structures because such data sets are primarily sourced from large redundant databases such as PDB[78] (see Table S2). However, clustering in the sequence space can be insufficient in some cases. For example, distantly related proteins may have very similar active site geometries even though their sequence homology is low.[79] Clustering may thus be performed at the level of the structural representation instead. While such strategies have only rarely been used in the past, they may become more common due to the emergence of new tools for protein structure searching such as Foldseek.[80]

3. PROTEIN ENGINEERING TASKS SOLVED BY MACHINE LEARNING

This section provides a brief overview of the protein engineering tasks that have already drawn the attention of machine learning developers. The first of these tasks is functional annotation, which is important because the overwhelming majority of sequences in protein databases remain unannotated and *in silico* prediction is necessary to keep pace with the exponentially growing number of deposited sequences. We will then cover the available labeled data sets and state-of-the-art ML tools for predicting mutational effects in proteins, as well as strategies for protein design based on their predictions. In addition, we will review published methods for leveraging unlabeled protein sequence and

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

structure data sets to help guide protein engineering. Finally, we conclude with examples showing how ML models of protein dynamics can facilitate the selection of promising mutations.

3.1. Functional Annotation of Proteins. Knowledge of protein functions is fundamental for protein engineering pipelines. For instance, in protein fitness optimization, scientists start from a characterized wild-type sequence with some degree of the desired function.[81−83] Likewise, in biocatalysis, one needs to know the function of enzymes to assemble a biosynthetic pathway from plausible enzymatic reactions.[84,85]

Traditionally, scientists have characterized the functions of proteins via laborious, time-consuming, and costly wet lab experiments. However, owing to high-throughput DNA sequencing, the exponentially growing repertoire of protein sequences is reaching numbers far beyond the capabilities of experimental functional annotation; for example, the Big Fantastic Database (BFD, Table S2) contains 2.5 billion sequences to date. Functional annotation is particularly important for enzymes. A broad-level annotation (*e.g.*, enzyme family) can be achieved relatively easily by sequence homology and by searching for protein domain motifs, but detailed annotation of enzyme substrates and products currently requires experimental characterization. To accelerate this process, a substantial recent effort has been dedicated to the development of novel computational methods for functional annotation.

The developed computational methods heavily rely on data sets of previously characterized enzymatic functions for training. For example, information on protein families and domains is often sourced from the Pfam,[86] SUPERFAMILY,[87] or CATH[88] databases. Enzymatic activity data often come from databases such as Rhea,[89] BRENDA,[90] SABIO-RK,[91] PathBank,[92] ATLAS,[93] and MetaNetX.[94] The ENZYME database under the Expasy infrastructure[95] provides the Enzyme Commission (EC) numbers, the most commonly used nomenclature for enzymatic functions. EC numbers are a hierarchical classification system categorizing enzymatic reactions at four levels of detail, with the fourth level being the most detailed. An EC number groups together proteins with the same enzymatic activity regardless of the reaction mechanisms.[96] The data from the above-listed databases have also been post-processed and organized into data sets such as ECREACT[97] or EnzymeMap,[98] which should further facilitate the development of computational models.

The models for enzymatic activity prediction typically use enzyme amino acid sequences as input, as the goal is to directly annotate the outputs of high-throughput DNA sequencing. The incorporation of the structural inputs, facilitated by the recent breakthroughs in structure prediction,[67,99] is still to be explored more by the community. The outputs of these models generally fall into three categories based on the resolution of predictions. First, the most general models predict protein families and domains. Second, the EC class-predictive models provide a more detailed estimation of enzymatic activity. Finally, the most comprehensive picture of enzymatic activity requires models for predicting an enzyme's substrates and corresponding products. Several recent deep learning models predict protein families and domains.[100,101] Although such models are crucial for studying proteins, they have only a limited applicability in enzymatic activity prediction, as a single protein family can combine enzymes catalyzing different reactions.[102] To meet the needs of protein engineering, the models predicting the EC numbers appear to be more relevant, as they can capture the catalytic activities.

Over the years, the community attempted to predict EC numbers using multiple sequence alignment and position-specific scoring matrices (PSSM) or hidden Markov model (HMM) profiles,[103−105] *k*-nearest neighbor-based classifiers,[104−108] support vector machines (SVM),[105,109−112] random forests,[113,114] and deep learning.[60,115−119] Most methods approached the prediction of EC numbers as a classification problem, which led to poor performance on the under-represented EC categories. The recent deep-learning-based method[119] tackled the EC number prediction via a contrastive learning approach, training a Siamese neural network on top of sequence **embeddings** from a pretrained protein language model.[66] The resulting predictive algorithm, CLEAN, can better identify enzyme sequences that belong to any EC category, including underrepresented ones. CLEAN achieved state-of-the-art performance in EC number prediction *in silico*, and it was experimentally validated *in vitro* using high-performance liquid chromatography−mass spectrometry coupled with enzyme kinetic analysis on a set of previously misannotated halogenase enzyme sequences.

EC class prediction enables downstream applications such as retrobiosynthesis.[85] For instance, planning of biosynthesis has been tackled by predicting the chemical structure of the substrate and the required enzyme EC number from the provided enzymatic product using a transformer-based neural network.[97] Several other published deep learning models aspired to estimate the substrate of an enzymatic reaction based on its product.[120,121] Such models can be used to prioritize enzyme selection based on the EC class or the substrate/product pair. However, the assignment of a specific enzyme sequence (*i.e.*, not only the EC class) to a desired reaction remains a challenge for future development.

Recently, the first general models predicting interaction between individual enzymes and substrates/products were published.[122,123] These DL-based models take pairs of a protein sequence and a small molecule as inputs and predict their possible interaction. Unfortunately, none of the general substrate−enzyme interaction models were validated in wet lab experiments. Furthermore, in Kroll et al. the authors admit poor generalization of the model to out-of-sample substrates.[123] Moreover, enzyme-family-specific models were shown to outperform the general models in predicting the enzyme−substrate interactions.[124] To sum up, the practical applicability of general enzyme−substrate interaction models has yet to be determined.

3.2. Supervised Learning to Predict the Effects of Mutations. The ability to predict mutational effects on various protein properties, such as solubility, stability, aggregation, function, and enantioselectivity, is another desirable goal of protein engineering. From the machine learning perspective, this implies having a model that takes a reference protein and its variant as the input and predicts the change in the studied property as the output. Intuitively, this could be achieved by supervised learning on the labeled data sets of wild-type proteins first (*e.g.*, to predict a solubility score, binding energy, or melting temperature of a given protein), applying the trained model to independently predict labels of the reference protein and its variant, and then taking the difference between the two predicted scores. The attractiveness of this strategy comes from large annotated data sets of wild-

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

Table 2. Recent Applications of Supervised ML Tools in Prediction of the Effects of Mutations in Protein Engineering[a]

targeted property	tool	training data	method	input	Web site
stability (ΔΔG)	BayeStab [134]	2648 single mutations from 131 proteins (S2648) derived from ProTherm [135]	Bayesian neural networks	PDB files with the WT and mutant structures	http://www.bayestab.com/
	PROST [136]	2647 single mutations from 130 proteins (S2648) derived from ProTherm [135]	ensemble model	sequence (FASTA) plus a list of mutations	https://github.com/ShahidIqb/PROST
	KORPM [137]	2371 single mutations from 129 proteins, derived from ThermoMutDB [138] and ProTherm [135]	nonlinear regression	a list of PDB files and single-point mutations	https://github.com/chaconlab/korpm
	ABYSSAL [139]	376 918 single mutations from 396 proteins [140]	siamese deep neural networks trained on ESM2[67] embeddings	ESM2 embeddings of the WT and mutant sequences	https://github.com/dohlee/abyssal-pytorch
solubility	PON-Sol2 [141]	custom data set: 6328 single mutations from 77 proteins	LightGBM	a list of sequences (FASTA) plus lists of single-point mutations	http://139.196.42.166:8010/PON-Sol2/
activity	DLKcat [142]	custom data set: 16 838 data points from BRENDA [90] and Sabio-RK [91]	graph-based and convolutional neural networks	a list of substrate SMILES and enzyme sequences	https://github.com/SysBioChalmers/DLKcat
	MaxEnt [143]	various custom MSAs and kinetic constant data sets	statistical Potts model	single and pairwise amino acid frequencies from MSA	https://github.com/Wenjun-Xie/MEME
	MutCompute [31]	19 436 protein structures	self-supervised convolutional neural network	protein structure	https://mutcompute.com/
	innov'SAR [144]	custom data set: 7 variants vs 3 substrates at two pH levels	partial least-squares regression	N/A	N/A
	ML-variants-Hoie-et-al[145]	custom data set: over 150 000 variants in 29 proteins	random forest	Custom preprocessing, derived from PRISM approach [146]	https://zenodo.org/record/5647208 https://github.com/KULL-Centre/papers/tree/main/2021/ML-variants-Hoie-et-al
optimal catalytic temperature T_{opt}	ML-guided directed evolution of a PETase [147]	custom data set: 2643 enzymes from BRENDA[90]	random forest	N/A	N/A
	TOMER [148]	custom data set: 2917 enzymes from BRENDA[90]	bagging with resampling	a list of sequences (FASTA)	https://github.com/jafetgado/tomer
substrate specificity	ML-guided directed evolution of an aldolase[149]	131 experimentally characterized variants	Gaussian process	N/A	N/A
protein aggregation	ANuPP[150]	1421 hexapeptides obtained from CPAD 2.0,[151,152] WALTZ-DB,[153,154] and AmyLoad[155] databases	ensemble of 9 logistic regressors	a list of sequences (FASTA)	https://web.iitm.ac.in/bioinfo2/ANuPP/about/
	AggreProt	1416 hexapeptides obtained from WALTZ-DB[153,154]	deep neural network	up to 3 sequences (FASTA) and (optionally) 3D structures	https://loschmidt.chemi.muni.cz/aggreprot
binding affinity (ΔΔG)	GeoPPI [156]	SKEMPI 2.0 [157]	graph neural network and random forest	PDB file, the names of interacting chains, and mutations	https://github.com/Liuxg16/GeoPPI

[a]N/A = not available.

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

type proteins available for training. For example, the Protein Structure Initiative[125] generated a massive data set TargetTrack, often used for protein solubility prediction.[26] Additionally, the more recent Meltome Atlas of protein stability obtained by liquid chromatography-tandem mass spectrometry[126] was used for predicting melting temperatures.[127] Our ongoing effort to predict highly valuable melting temperatures solely from the protein sequence resulted in the development of the TmProt software tool (https://loschmidt.chemi.muni.cz/tmprot/). Large labeled data sets usually provide enough training data for powerful end-to-end deep learning.[34,128,129] Nonetheless, when the training data set does not contain mutations, a few substitutions will usually result in similar predicted labels (*e.g.*, solubility scores), in contrast to dramatic changes often observed in experiments. Therefore, the strategy of taking the difference between the predicted labels for a reference protein and its variant typically fails to produce reliable predictors for mutational effects.[130]

A more promising route is to use labeled mutational data sets for training. This strategy has its own limitations, since such data sets are not only scarce but also sparse in terms of the extent of the mutational landscape that is probed (the sequence space grows exponentially with the number of mutated residues) and biased toward several overrepresented proteins.[13,131] These barriers severely hinder the use of ML, which relies heavily on the availability of good quality data with a high coverage of the space of interest. Therefore, additional data curation and processing, adjustments to training protocols, and more thorough and critical data evaluations are typically needed. Such efforts will be a crucial first step in establishing reliable ML pipelines for predicting mutational effects.

The most abundant and diverse mutational data come from general biophysical characterizations that are performed routinely in most protein engineering studies, including measurements of protein expressibility, solubility, and stability. The major challenge when using such data lies in collection and curation: measurements are scattered across the literature and often reported *ad hoc* because they are generally complementary to a study's main results.[132] This highlights the importance of establishing and maintaining databases with protein annotations to facilitate data discoverability and reuse. For example, we recently released SoluProtMutDB,[133] which currently has almost 33 000 labeled entries concerning mutational effects on the solubility and expression of over 100 proteins. This database incorporates all of the data points that were recently used to develop solubility predictors (Table 2), which achieved correct prediction ratios of around 70%.[133]

Protein stability measurements are another type of widely available biophysical data that can be used in ML. Protein stability is typically quantified in terms of the melting temperature (T_m) or the Gibbs free energy difference between the folded and unfolded states ($\Delta\Delta G$). Several protein stability databases exist, including FireProtDB,[158] ThermoMutDB,[138] and ProThermDB,[135] and their data have often been used to train ML predictors that have achieved Pearson's correlations of up to 0.6 and RMSE values of 1.5 kcal/mol when applied to independent test sets.[159] Interestingly, these numbers have barely changed over the past decade, indicating that a qualitative paradigm shift might be needed to advance ML-based prediction of mutation-induced protein stability changes.[132] It is possible that large new data sets could provide the necessary boost, and some exciting studies collecting such data are already appearing: cDNA display proteolysis was recently used to measure the thermodynamic stability of around 850 000 single-point and selected double-point mutants of 354 natural and 188 *de novo* designed protein domains between 40 and 72 amino acids in length.[140]

Changes in catalytic activity upon mutation also attract the attention of ML researchers. Predicting mutational effects on enzyme activity is more challenging than predicting protein stability and solubility due to the enormous diversity of enzymatic mechanisms. One rich source of such mutational data is large-scale deep mutational scanning.[69] These experiments combine high-throughput screening and sequencing and typically score protein variants by comparing their abundance before and after a specific selection is applied. These data sets provide comprehensive overviews of the local mutational landscapes of various enzymes and are of significant value for ML due to their unbiased mutant coverage. Several groups have already assembled various deep mutational scanning (DMS) data sets for benchmarking effect predictors,[145,160−164] and we expect this trend to continue as more data sets appear. Notable works of this type include the recently published activity landscapes of the phosphatase,[165] dihydrofolate reductase,[166] DNA polymerase,[167] and palmitoylethanolamide transferase.[168]

DMS can be applied to a wide range of enzyme functions due to its high flexibility with respect to selection procedures. However, its high throughput comes at the cost of limiting the number of protein targets that can be used in a study; often, only a single case is examined. The desire to target multiple enzymes simultaneously motivated the creation of another notable database of enzyme activity changes: D3DistalMutation.[169] It contains data derived from UniProt annotations representing over 90 000 mutational effects in 2130 enzymes. However, its potential in ML has not yet been explored.

Other protein characteristics may also be used as targets for protein engineering and machine learning. These targets are often selected based on the enzyme of interest and may include important functional traits such as substrate specificity,[149] enzyme enantioselectivity,[170,171] kinetic constants,[142−144] temperature sensitivity,[172] or temperature optima.[147,148] In addition, several mutational data sets that can be used for ML-based tools focusing on protein folding rates, binding, and aggregation have been deposited in the VariBench benchmark data set.[173] Selected recent examples of these tools are listed in Table 2. An overview of the described databases and data sets is given in Table S2.

3.3. Approaches to Design Mutations. While tools for predicting effects of mutations have become increasingly advanced in recent years, in their simple form, they can only provide labels for a given substitution. However, the desired outcome of protein engineering pipelines is to have a list of promising protein variants for experimental validation. Therefore, even if a reliable ML-based tool for predicting effects of substitutions is available, the problem of suggesting promising hypothetical designs must be addressed. This problem may become a major bottleneck, since even if the prediction of single- or multiple-point mutational effects is fast, evaluating all possible combinations of mutations remains unfeasible. Therefore, there is a growing need for tools that simultaneously predict the effects of mutations and reduce the search space, which is the focus of this subsection.

A major challenge in the development of such tools is finding ways to efficiently reduce the space of multipoint

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

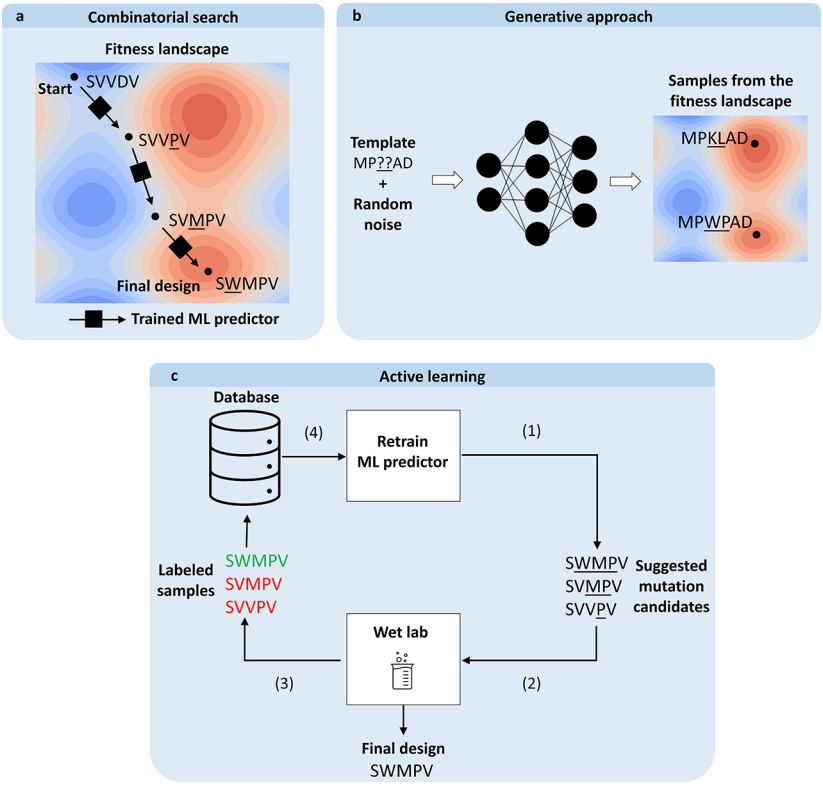


Figure 3. Selected ML strategies for designing new enzyme variants. (a) *In silico* evolutionary combinatorial search for favorable mutations. A machine learning predictor is used to iteratively evaluate candidate mutations for some desired property, *e.g.*, stability. The mutated amino acid in each step is underlined. (b) Generation of favorable mutants. The machine learning tool directly infers the sequence with high property values (*e.g.*, stability) from a fitness landscape learned and captured in the weights of the model during the training. (c) An active learning loop: an ML predictor from (a) or (b) is used to (1) propose enzyme variants that are (2) evaluated experimentally, and the resulting data are used to (3) update the knowledge database. The ML model is then (4) retrained on the updated knowledge database, and new variants are designed.

mutants. In an analysis of nine case studies, Milton and coauthors found that the effects of half of the multipoint mutations influencing enzymatic properties could not be predicted using knowledge of the corresponding single-point mutations, with the associated complexities resulting from direct interactions between residues in some cases and long-range interactions in others.[174] This common nonadditive behavior, which is known as epistasis, has prompted the development of ML models and combinatorial optimization algorithms capable of scoring or searching multipoint mutants by design. Several approaches discussed below have been proposed to overcome this challenge, more on this topic can be found in a review.[175]

One such approach is to produce a library of variants for screening using reliable physics- and evolution-based tools. Even a time-consuming preselection of promising hotspots can drastically reduce the computational time of the downstream ML scoring and search.[176] For example, HotSpot Wizard 3.0[177] achieves robust selection of hotspots by using a number of sequence and structure-based filters to identify mutable residues for which mutation effects are then quantified using the well-established Rosetta and FoldX tools. Another example is the FuncLib web server, which computes promising single-point active-site mutations using evolutionary conservation analysis and Rosetta-based stability calculations.[178] This tool exhaustively models each combination of mutants and ranks them by energy. Evolutionary information can also be captured by ML-based models[179] and used to suggest promising substitutions, *e.g.*, amino acids with conditional likelihoods higher than the wild-type.[180]

Mathematical optimization methods can generate promising protein sequence candidates *in silico* by iteratively producing new designs based on available ML scoring data (Figure 3). One group of such methods uses the ML predictor as a black-box oracle to evaluate existing candidates. This evaluation is then used to approximate the "fitness" of sequences, which is in turn used to navigate the sequence landscape and generate a new set of candidates using tools such as evolutionary algorithms[82,181] or simulated annealing.[182] However, approximating complex mutational landscapes using oracles representing estimated, simplifying distributions can harm the optimization process and prevent the optimal solution from being found.[183] Adaptive sampling of the design space can be used instead to obtain better results.[81] Other alternatives are to use generative models[184] or rely on so-called white-box optimization, which involves using knowledge of a predictor's internal workings to find the optimal solution. For example, linear regression coefficients can be used to suggest modifications (mutations) of the input that alter the corresponding features in the desired direction. White-box methods are discussed further in the context of **explainable AI** in section 6.1.

While *in silico* optimization methods enable the iterative generation of promising designs, they rely entirely on the ability of a predictor to correctly score any given point in a mutational landscape, which might be an unrealistically strong assumption. A possible alternative (albeit one that is more

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

Table 3. Some Recent Self-Supervised ML Models for Protein Engineering[a]

approach	tool	training data	method	Web site
large sequence-based models	ESM-2 [67]	∼65 million unique sequences sampled from UniRef50 [203]	transformers, 15 billion parameters, masked (15%) language modeling	https://github.com/facebookresearch/esm
	ProtTrans [197]	2122 million sequences from Big Fantastic Database used for pretraining and fine-tuning on 45 million sequences from UniRef50 [203]	transformers, 11 billion parameters, masked (15%) language processing	https://github.com/agemagician/ProtTrans
	Ankh [199]	45 million sequences from UniRef50 [203]	transformer, 1.15 billion parameters, 1-gram span partial demasking/reconstruction (20%)	https://github.com/agemagician/Ankh
	ProGen[35,204]	281 million nonredundant protein sequences from > 19 000 Pfam[205] families	transformer-based conditional language model, 1.2-billion parameters	https://github.com/salesforce/progen
	Tranception [164]	249 million sequences from UniRef100 [203]	autoregressive transformer architecture, 700 million parameters	https://github.com/OATML-Markslab/Tranception
large structure-based models	MutCompute [31]	19 436 nonredundant structures with resolution better than 2.5 Å drawn from structures in the PDB-REDO database[206]	3D convolutional neural network, amino acid local environment separated into biophysical channels	https://mutcompute.com/
	ProteinMPNN [207]	19 700 single-chain protein structures,[78] with resolution better than 3.5 Å and <10 000 residues in length	message-passing neural network, 1.6 billion parameters	https://huggingface.co/spaces/simonduerr/ProteinMPNN https://github.com/dauparas/ProteinMPNN
	FoldingDiff [27]	24 316 training backbones from the CATH data set[208]	denoising diffusion probabilistic model	https://github.com/microsoft/foldingdiff
	RFdiffusion [28]	structures sampled from PDB[78]	denoising diffusion probabilistic model	https://github.com/RosettaCommons/RFdiffusion
	GearNet [68]	805 000 predicted protein structures from AlphaFoldDB[209]	graph neural network, 42 million parameters	https://github.com/DeepGraphLearning/GearNet
	Stability Oracle [33]	22 759 PDB structures[78] with resolution better than 3 Å, maximum 50% sequence similarity, and ∼144 000 free energy labeled data points for fine-tuning	graph transformer, 1.2 million parameters	N/A
models for specific protein families	MSA Transformer [210]	26 million MSAs with 1192 sequences per MSA on average	transformers, 100 million parameters, axial attention [211]	https://github.com/rmrao/msa-transformer
	ProteinGAN [212]	16 706 sequences of bacterial malate dehydrogenase	generative adversarial networks, 60 million parameters	https://github.com/biomatterdesigns/ProteinGAN
hybrid methods	Prot-VAE [213]	46 million sequences from UniRef sequences and ∼5300 SH3 (Src homology 3 family) homologues	transformers, 1D convolutional neural network and a custom architecture	N/A
	ECNet [214]	protein sequences from Pfam[205]	transformers with a 38 million parameter model, bidirectional LSTM[215]	https://github.com/luoyunan/ECNet
	ESM-GearNet [32]	805 000 predicted protein structures from AlphaFoldDB[209] and ∼65 million unique sequences sampled from UniRef50 [203]	transformers, 15 billion parameters, masked (15%) language modeling (sequences), graph neural network, 42 million parameters (structures)	https://github.com/DeepGraphLearning/GearNet

[a]N/A = not available

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

costly and has lower throughput) is to directly incorporate experimental validation into the optimization loop. This experimental input can guide search algorithms to more promising parts of the mutational landscape in a manner that is akin to directed evolution. While advanced search methods of this type have previously been used to improve traditional directed evolution,[82,182] they do not fully exploit the potential of experimentally characterizing intermediate variants. ML-based **active learning** methods accelerate directed evolution by iteratively extracting knowledge from all characterized variants and selecting the most promising ones.[185] By relying on the new experimental data, only a limited number of training samples can be expected,[186] confining the choice of ML models to those with a lower number of parameters, such as **multilayer perceptrons** (MLP) with as few as two layers.[187] A recent advance in the area of active learning is the development of GFlowNets,[188] networks designed to suggest diverse and accurate candidates in a machine−expert loop to accelerate scientific discovery. Studies using such networks have demonstrated their potential for designing small molecules;[189] however, the utility for the design of proteins remains to be reliably demonstrated.

Powerful design techniques for the exploration of variants of enzymes and other proteins often rely not only on combining experimental and *in silico* techniques but also on combining multiple *in silico* approaches and sometimes different ML techniques. For example, focused training ML-assisted directed evolution (ftMLDE)[186] combines unsupervised and supervised training approaches by using unsupervised clustering to construct a training set for supervised classifiers. These classifiers are then used to select promising mutants with tools such as CLADE 2.0.[190] Similarly, unsupervised learning on millions of sequences was used to obtain a protein representation called UniRep,[191] which was further tuned in an unsupervised manner to obtained eUniRep (evotuned UniRep) for proteins related to the target sequence, leading to an informative set of features for proteins in general as well as for the target in particular. Such representation enabled data-efficient supervised learning of a model for guiding *in silico* evolution.[192] Alternatively, an unsupervised "probability density model" has been used to produce an "evolutionary density score", a feature that was then used to augment a small number of labeled data points on which a light model was trained in a supervised manner. Interestingly, such an approach was shown to outperform the supervised **fine-tuning** of the probability density model pretrained in an unsupervised manner.[193] Another approach[194] combines self-supervised large protein language models with a supervised structure-to-sequence predictor in a new and more general framework called LM-design that is claimed to advance the state of the art in predicting a protein sequence corresponding to a starting backbone structure, sometimes called "inverse folding". While inverse folding does not explicitly search the mutational landscape, it can be used to identify promising mutations by inputting an existing protein structure and a partially masked sequence and using the inverse folding tool to propose amino acids for the masked parts.

3.4. Leveraging Unlabeled Data Sets to Score Mutations. Over the past decade, **large language models** (LLM) have become popular tools for solving NLP problems ranging from language translation to sentiment analysis.[195] This major paradigm shift was driven by the realization that even unlabeled data contain useful information: the distributional hypothesis suggests that the meaning of words can be deduced by analyzing how often and with what partner words they appear in various texts.[196] Analogously, in biology, we can regard proteins as sequences based on "the grammar of the language of life", which implies that the distribution of amino acids at specific locations can provide valuable insights that could be used to help predict the effects of substitutions on protein function, thereby reducing the reliance on external data sources.[66] For example, Elnaggar et al. showed that the embeddings generated by a LLM, when used as input features, can effectively facilitate the development of small supervised models whose predictive power rivals that of state-of-the-art methods relying on evolutionary information obtained from MSAs.[197] Additionally, the protein language model ESM-2 was recently trained on protein sequences from the UniRef database to predict 15% of masked amino acids in a given sequence.[67] This made it possible to directly leverage sequence information to greatly improve B-cell epitope prediction[198] without a supplementary MSA. The inherent attention mechanism of ESM-2 can also be used to facilitate protein structure prediction.[67] We provide more examples in Table 3. Furthermore, the embeddings obtained from the language models, such as ESM-1b,[65] ESM-2,[67] ProtT5,[197] and ProtTrans,[199] have become a popular way of representing sequential data, making pretrained ESM models a frequent subject for **transfer learning**. The transfer of knowledge from these models has been tackled by fine-tuning the pretrained weights,[197,200] by training models solely on top of the learned embeddings (keeping the pretrained weights fixed),[197] and also by introducing adapter modules[201] between the trained layers for parameter-efficient fine-tuning.[202] The power of the learned embeddings can also be witnessed in a completely unsupervised setting. More specifically, the ESM-1v[66] model was demonstrated to accurately score protein variants by relying solely on the wild-type amino acid probabilities learned by the protein language model during pretraining without any subsequent fine-tuning.

Large deep-learning models are impressively capable of learning general protein properties. For the cases in which a detailed understanding of a specific protein or protein family is required rather than general patterns that hold across different protein families, sequence-based models that can learn distributional patterns from a single MSA are available. MSAs have already proven to be a rich source of evolutionary information, *e.g.*, for identifying functionally important conserved regions, insertions, or deletions to clarify the mechanisms driving sequence divergence.

Analyses of evolutionary data and comparisons of assigned probabilities for mutant and wild-type sequences have also shown that ML-based models can predict the effects of mutations in deep mutational scanning experiments more accurately than established methods based on evolutionary data.[66] For example, excellent protein structure prediction performance was achieved by combining MSA inputs with an advanced transformer architecture.[99,210,216] In addition, Xie et al. used the maximum entropy (MaxEnt) principle to infer statistical energy from homologous sequences[143] and found that the inferred statistical energy of active site residues correlated significantly with enzyme activity, whereas that of residues distant from the active site correlated with protein stability. Hsu et al. observed that a hybrid linear regression model combining the evolutionary density score from MSA-based ML models with labeled one-hot encoded protein

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

sequence data sets demonstrated superior performance in a range of protein fitness prediction tasks, even for the sizes of the labeled data sets in the range of 48−240 data points.[193] A similar effect for small data sets of 50−250 data points was reported by Illig et al.[217] Ding et al. showed that variational autoencoders can capture phylogenetic relationships in the geometry of the latent space of an MSA and further demonstrated that the free energies of sequences assigned by the latent model can be used to predict mutation-induced changes in stability.[218] Building on these findings, the geometric structure of a latent space was recently used to guide the design of a haloalkane dehalogenase.[219] In addition, by exploring the latent manifold underlying the sequence information, we can uncover dependencies that may not be readily apparent in the raw latent space embeddings.[220]

Despite its advantages, MSAs also have some drawbacks. First, it can be difficult to create an MSA that contains enough evolutionarily relevant sequences to establish strong patterns at key amino acid positions. Second, the creation of an MSA is often seen as something of a craft rather than a systematic procedure, and the alignment process can be sensitive to user choices, including the choice of substitution matrices, gap penalties, and the number of iterations. Poor choices may produce an MSA with improperly aligned residues because too few iterations were performed during the alignment process or because a significant number of gaps were introduced (particularly at the beginning and end of the alignment), necessitating additional preprocessing and trimming of the MSA to achieve optimal results. Moreover, aligning new sequences to an MSA can be a challenging task that requires careful consideration of the alignment parameters and sequence properties. Despite these challenges, MSAs remain a powerful source of information for studying the relationships and evolutionary history of biological sequences.

Recent studies have also explored the opportunities provided by examining patterns at two contrasting scales within the protein sequence space: general patterns in the large universe of protein sequences and local distributions of sequence patterns in specific protein families. These works have combined large pretrained models with lightweight, easily retrainable components for efficient family-specific adaptation. For example, Sevgen et al. designed ProtT-VAE, a fusion of a transformer pretrained on 46 million UniRef50 sequences and an autoencoder that enables fine-tuning on a sequence library of interest.[213] After fine-tuning the model on the phenylalanine hydroxylase (PAH) sequence family, ProtT-VAE was used to predict variants with up to 100 mutations, resulting in a 2.5× increase in catalytic activity over the human wild-type. Similarly, Luo et al. combined a pretrained language model with an MSA-specific direct coupling method to capture both general protein syntax and protein-specific epistasis.[214] The resulting model, ECNet, was used to engineer TEM-1 β-lactamase variants with improved ampicillin resistance.

The utilization of unlabeled data in a semisupervised setting has been tried, for example, in the context of secondary structure prediction for orphan sequences[221] or structural similarity prediction for protein sequences.[222] These two methods employ pseudolabeling or a custom similarity metric to enable a supervised learning task to profit from large initially unlabeled data. In some cases, even self-supervised methods are viewed as semisupervised[223] in that they learn powerful representations from unlabeled data that can then be fine-tuned using small labeled data sets. However, the difference from semisupervised methods is that self-supervised methods build on the established methodology of supervised learning as they formulate a supervised proxy task using synthetic labels generated automatically from the large unlabeled data set. Such self-supervised methods are powering the recent successes of protein language models, discussed at the beginning of this section, and appear to be generally more successful than traditional semisupervised methods.

In parallel to the explosive growth of sequence databases, the Protein Data Bank reached the landmark value of 200 000 entries in April of 2023, providing a rich source of experimental structural data for self-supervised learning about protein structures. A number of deep learning models have been fitted to large subsets of the PDB data to leverage the natural diversity of the protein structures. For example, MutCompute was trained on a nonredundant sample of 19K proteins to predict artificially masked residues based on the local 3D environment, enabling the model to successfully capture phenotypic landscapes associated with protein stability.[31] Additionally, Zhang et al. designed multiple general tasks for the self-supervised pretraining of graph neural networks for protein structures, leading to improved performance in various downstream tasks.[68] Self-supervised learning on protein structures has also been used to suggest protein sequences for specific backbones and generate *de novo* protein structures.[27,28,207]

3.5. Leveraging Protein Dynamics to Compare Mutations. The preceding sections focus on the analysis of static data. However, proteins are complex and dynamic systems, and their conformational changes and motions often provide key insights into reaction mechanisms that cannot be obtained by studying static structures alone.[224] One way to capture a protein's dynamics is by studying its structural ensembles, which are available from existing protein structure databases in many cases. It has therefore been suggested that structure-based ML methods trained on such data, *e.g.*, AlphaFold2, could provide insights into protein dynamics.[225] Building on this hypothesis, Brotzakis and coauthors proposed a reweighting procedure using AlphaFold2 predictions and the FoldingDiff framework[27] to generate structural ensembles for disordered peptides.[226] Such procedures could be useful alternatives to computationally expensive simulations in some cases. However, they are unlikely to be useful for comparing structural ensembles of closely related protein variants because AlphaFold2 predictions, which rely on evolutionary information obtained from MSAs, appear to be insensitive to single-point mutations[139] and so may be unable to accurately capture the often subtle differences between closely related protein variants.

Another option is to study molecular dynamics (MD) trajectories. MD data are obtained by performing simulations that apply physical laws to calculate the future position of every atom in a protein after a given time step based on its 3D structure at the current point in time. The resulting trajectories consist of a series of snapshots capturing the protein configurations at successive points in time. Because each snapshot contains the coordinates of all atoms in the simulated system, the simulation provides a large amount of data, even for an average-sized protein. Where prior system knowledge is available, one can use a coarse-grained model to reduce the system's dimensionality and potentially capture major motions of interest.[227] In most cases, however, finding so-called collective variables (CVs) that effectively describe the system's

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

dynamic behavior is not straightforward. It has been suggested that unsupervised learning methods could help solve the problem of identifying CVs because they can learn from raw MD trajectories without making assumptions about the CVs that are being sought. In general, such methods try to find low-dimensional representations that are rich enough to reconstruct the original (or time-shifted) input.[228] The resulting low-dimensional projections can then be used to build a simplified model of the protein's dynamics. This is often done using Markov state models (MSMs), which can cluster the conformational space of the simulated molecule into a tractable number of clusters (states). MSMs assume that the transitions between these states are Markovian, *i.e.*, the state in the next time step depends only on the current state regardless of the previous trajectory. This approach was applied in conjunction with the end-to-end deep learning model VAMPnet, which was used to find an optimal projection for the system under study.[229] More recently, the CoVAMPnet framework[230] was created by expanding on the VAMPnet approach to add interpretability capabilities and a method for MSM alignment that facilitates comparisons of MSMs for two sets of simulations. If applied to different variants of a given protein, CoVAMPnet could potentially be used to evaluate the effects of discriminative mutations on protein dynamics. However, this application is yet to be explored.

In general, identifying dynamic features associated with biochemical differences between protein variants is much more challenging than finding a low-dimensional representation for a single protein. One way to address this problem was demonstrated by the DiffNets framework,[231] in which supervised autoencoders were trained on MD trajectories to identify the most significant differences in pairwise residue distances between protein variants. Another approach for comparing dynamical changes in variant trajectories involves directly analyzing the distribution of low-dimensional projections of configurations obtained by variational autoencoders. In this approach, similar spatial configurations tend to cluster in certain subspaces of the learned low-dimensional space. Such clustering can be used, for example, to analyze the MD simulations of catalytic sites in the presence of different substrates to identify structural differences that drive substrate preferences.[232] Nonetheless, despite those promising studies, the problem of comparing MD trajectories of different protein variants in a systematic way remains largely unexplored, offering an intriguing future direction for machine learning applications.

4. RECENT SUCCESS STORIES AND LESSONS LEARNED

While the number of publications on ML for protein engineering is growing rapidly (Figure 1), only a fraction of these publications incorporate the experimental validation of generated predictions as opposed to validation using existing data. This section highlights several recent publications that do include experimental validation and clearly demonstrate the great potential of ML in enzymology. We provide more examples of such studies in Table S1.

4.1. Sequence-Based Case Studies. Several publications have showcased recent advances in the use of sequence-based models, including both models trained on large sequence databases and models trained on specific enzyme subfamilies. For example, Russ et al. used a statistical model based on direct coupling analysis to learn the natural distribution of amino acids in chorismate mutases and generate 1618 artificial enzyme sequences. Experimental studies showed that 30% of these proteins rescued the growth of enzyme-deficient *Escherichia coli* in minimal media.[233] Ten of them were characterized in more detail and shown to have expression and catalytic parameters similar to those of the previously characterized enzymes. In another case, Repecka et al. used generative adversarial networks to create artificial malate dehydrogenases.[212] Sixty of the resulting artificial proteins (which had pairwise sequence identities of 45−98% with natural enzymes) were tested experimentally, revealing that 13 had catalytic activity comparable to that of natural enzymes. Another interesting case study was reported by Madani et al., who used a large 1.2 billion-parameter language model called ProGen for the conditional generation of 100 artificial sequences that were fine-tuned to five distinct lysozyme families.[35] Again, the artificial designs had expression levels and catalytic efficiencies similar to those of natural lysozymes even though their sequence identities with the natural proteins were as low as 31.4% in some cases. In all three studies, the authors could approach the expression and activity of natural sequences, but surpassing them by a significant margin still remains a challenge.

4.2. Structure-Based Case Studies. Designs superior to wild-type templates could potentially be created by leveraging structural data. One promising strategy for this purpose is to use the local structural environment to identify positions suitable for optimization in wild-type proteins. An insightful recent study on plastic-degrading enzymes demonstrated the power of this approach.[234] Despite extensive research on enzymatic PET depolymerization, most known PET-hydrolyzing enzymes have poor activity and require high temperatures or highly processed substrates to be practically useful. Traditional protein engineering strategies have improved the thermostability and catalytic activity of PETase variants under certain conditions, but these variants still show low activity at mild temperatures. Lu et al. therefore tried to use structure-based deep learning to solve this problem.[234] For this purpose, they used the MutCompute[31] algorithm, which was trained to predict masked amino acids based on their local 3D microenvironment in a crystal structure, to identify the positions where the predicted probabilities of the wild-type amino acids were comparatively low, suggesting that some other amino acids are better "suited" to the corresponding structural microenvironment. Eight of the top ten suggested substitutions produced single-point mutants with improved thermostability and activity. Combinatorial assembly of these substitutions yielded PETase variants with melting temperatures of up to 83.5 °C (and thus greater thermal stability than any previously reported variant of this enzyme) and up to 38× the activity of the template enzymes at 50 °C.

MutCompute was also used to improve the thermostability of the Bst DNA polymerase, with a similar success rate: of the top 10 substitutions suggested by the network, only two were discarded for showing little or no activity, while the top five yielded activities equal or superior to the template.[235] Moreover, variants combining these substitutions were more robust to purification and exhibited even greater thermotolerance and thermostability. MutCompute predictions are orthogonal to force-field calculations and phylogenetic analyses conducted with the popular automated web tools PROSS[236] and FireProt,[237] and these predictions can be exploited to remove destabilizing substitutions from multiple-point mutants

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

designed using those tools. This was recently demonstrated by the successful stabilization of haloalkane dehalogenase DhaA115 (PDB ID 6SP5[238]).

An alternative to replacing specific residues to better match structural patterns in training data is to predict complete protein sequences corresponding to a given backbone conformation from scratch. Exemplifying this approach, Dauparas et al. trained the graph-based neural network ProteinMPNN on a set of 19 700 high-resolution single-chain structures from PDB and showed that this algorithm can rescue previously failed designs by suggesting optimized protein sequences for given scaffold templates.[207] For experimental validation, the authors targeted proteins that had been generated by deep network hallucination in a previous study[239] but proved to be mostly insoluble when expressed in *E. coli*. Ninety-six designs were processed using ProteinMPNN, of which 73 were soluble when expressed and 50 had the desired monomeric or oligomeric state. Moreover, some maintained this state even at 95 °C. This is a promising result because protein insolubility appears to be a persistent problem with ML-based generated sequences.[240] It will therefore be interesting to see if more enzyme sequences can be reengineered for improved solubility in this way.

At Loschmidt Laboratories, we developed the sequence-based solubility predictor SoluProt[26] and integrated it into EnzymeMiner (https://loschmidt.chemi.muni.cz/enzymeminer/), a more general pipeline for discovering enzymes with a desired catalytic activity that is available as a fully automated web service. This pipeline was recently used to mine promising industrially relevant haloalkane dehalogenases[241] and fluorinases.[242] In both cases, we obtained soluble and active enzymes with catalytic performance superior to that of previously characterized wild-type enzymes and engineered biocatalysts. The broad applicability of the SoluProt and EnzymeMiner web services is demonstrated by the fact that they have completed over 30 000 and 3000 jobs, respectively, in the two years that they have been online. An alternative to natural sequence mining with EnzymeMiner is to use the deep neural network ProteinMPNN, which has been applied successfully in the *de novo* design of artificial luciferases based on computationally designed binding pockets.[243] Functional constraints have also recently been incorporated into structure-based generative models including diffusion-based deep learning methods for functional motif scaffolding.[28]

4.3. Small-Data-Set-Based Case Studies. The preceding examples demonstrate the potential of using unlabeled data sets to suggest novel protein designs. Moreover, in section 3 we discussed strategies that leverage small data sets, for example, to fine-tune pretrained large models in order to enable more focused protein engineering. However, ML models trained using only small data sets with no pretraining can also be useful in protein engineering pipelines using simpler algorithms.

Several case studies combining machine learning and directed evolution have appeared recently. Based on the initial *in silico* docking, Büchler et al. chose three critical amino acid positions in an iron/α-ketoglutarate-dependent halogenase for full randomization in a library targeted for the use in an algorithm-aided enzyme engineering strategy.[244] After collecting the activity measurements for 504 unique variants, the authors trained a Gaussian-process-based model to explore *in silico* the remaining protein landscape for activity and selectivity. The subsequent experimental validation revealed active and selective halogenase variants with over a 90-fold increase in the apparent k_{cat} and a 300-fold increase in the total turnover number. A smaller data set of 80 variants of Sortase A was used by Saito et al. to train an ML model to score all possible variants for five mutated positions.[245] After designing primer sequences to include the top 50 variants in the second-round library, the authors observed most of the new variants showed high expression levels, with several demonstrating higher enzyme activity than the first-round variants. Reiterating the workflow, the authors constructed and validated the third-round library, again leading to a set of improved variants. Interestingly, the authors tested and stressed the importance of including poor-performing variants in the training data, and they still got promising results in the scenario in which the top-performing variant was excluded from training. Even a smaller initial library of 20 chimera sequences was used in a different study by Greenhalgh et al. as a starting point for ten rounds of sequence optimization of alcohol-forming fatty acyl reductases, leading to an over twofold increase in fatty alcohol production compared to the starting sequences.[246]

In all three case studies above, the authors used a set of simple sequence-based features such as one-hot encoding of amino acids, physicochemical properties of the proteins, or conservation scores. While extracting features from protein dynamics remains challenging (see section 5.2), a linear model was recently used to this end to elucidate structure−function relationships while engineering luciferases.[247] This study drew on an earlier indel (insertions and deletions) mutagenesis experiment targeting a reconstructed ancestral protein and aimed to identify the factors responsible for the emergence of dual dehalogenase and luciferase activities.[248] The authors comprehensively studied the dynamics of different variants and used partial least-squares regression to identify the strongest predictors of both activities. This knowledge was then used to obtain a design with lower product inhibition and highly stable glow-type bioluminescence.

These examples are notable because while few groups have the expertise or infrastructure needed for deep learning, simpler and more accessible ML methods can still be used to advance traditional protein engineering pipelines. At the same time, we expect that deep learning tools will gradually become more accessible and easier to use.

5. MAJOR GAPS IN THE STATE OF THE ART

Despite the exciting applications and promising case studies discussed above, several significant knowledge gaps remain to be addressed to take protein engineering to the next level. First, many protein engineering tasks have yet to benefit from ML, including predicting the effects of indels and unnatural amino acids, creating predictors that address several targets simultaneously, and predicting mutational effects on protein interactions. Second, molecular dynamics remains isolated from major advances in the application of ML to protein engineering; dynamical information is rarely if ever used when training current state-of-the-art predictors. Third, there is a pressing need to establish gold standard protein data sets because such benchmarks have significantly accelerated the progress of ML in other domains. Finally, the impact of ML-based tools often remains limited to a narrow circle of method developers, so there is a need to reach out to the broader community of biochemists and synthetic biologists. Below we discuss each of those gaps in more detail.

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

5.1. Unexplored Protein Engineering Tasks. One important objective for the future will be to create single ML tools that can perform multiple tasks. Multiobjective protein engineering is a core goal of many ongoing experimental studies because the introduction of mutations targeting one property often affects others. For example, stability and solubility are often negatively correlated and may also need to be traded off with other properties such as catalytic activity.[249] However, current ML-based predictors typically target only one property at a time. This limitation could potentially be overcome by combining predictions from multiple tools to define a so-called Pareto front, *i.e.*, a set of solutions in which no one member is better than another with respect to all objectives.[250,251] However, by combining the separately trained predictors, one misses the opportunity to train on a larger pool of data sets and potentially capture the common underlying mechanisms in a unified protocol. Approaches of this type are rarely used, possibly because their implementation would require expert knowledge of multiple types of data and the experimental techniques used to generate them.

Another major goal is predicting the effects of insertions and deletions in the amino acid sequence (indels). This area has been largely unexplored by ML even though indels occur frequently in nature and can unlock unique functional changes that substitutions alone cannot achieve.[252] Indel engineering is gaining momentum, however, and experimental studies focusing on indel mutagenesis are starting to produce interesting labeled data sets.[253,254] Protein evolutionary information is another potential source of data on indels[255] that can be used to explore previously hidden catalytic activities that could be shifted or promoted.[256,257] Indels can affect not only the structure of the protein system but also its dynamics.[258] Therefore, molecular dynamics simulations may provide valuable data for clarifying the effects of indels on protein dynamics and their functional consequences.

The vastness of the combinatorial protein sequence space can be further extended by introducing unnatural amino acid (UAA) substitutions. Experimental studies have successfully incorporated over 150 different UAA substitutions into protein sequences,[259] including multiple point mutants.[260] These substitutions can be used for diverse purposes, ranging from tailoring the structural, physical, and dynamic properties of specific sites to introducing new properties by adding carefully designed UAAs. Such additions can be used to enhance enzyme activity or elucidate the enzyme reaction mechanisms.[261] While there are some emerging ML-based tools for rational design of UAA sites,[262] further research and development are needed to make this approach widely applicable in enzyme engineering.

Another area that would benefit from the greater use of ML is the design of protein−protein and protein−ligand complexes. Despite recent promising results in the use of ML to predict the binding sites and protein−ligand complexes,[29,263,264] reliable and practically useful approaches for designing noncovalent interactions are still missing. For example, Geng et al. found that the evolution of ΔΔG predictors for protein−protein interactions had been hindered by the absence of centralized benchmarking.[265] Little progress has been made in this area, as evidenced by the re-emergence of similar ML models and the persisting lack of common evaluation standards.[156,266,267] Furthermore, the reliability of existing models is undermined by a frequent reliance on supervised ML with limited annotated data.[268]

Finally, most ML-based predictors represent primary protein sequences at the level of amino acids. However, synonymous mutations that do not change the amino acid sequence can still significantly impact protein expression and function[269,270] or can even relate to particular structural features.[271,272] Predictors working on the level of nucleotides or codons may thus be better tools for protein design and modification, particularly in areas such as prediction of expression, solubility, and aggregation. The fact that the 64-letter codon alphabet serves to encode richer information than the 20-letter amino acid alphabet can be directly exploited by ML models for improving performance on a wide range of tasks that are now being tackled at the protein sequence level.[273] While there have been several studies, *e.g.*, tackling protein expression optimization,[274] melting temperatures, subcellular localization, solubility or function,[273] we believe further research in this area might provide a strong boost for predicting many essential protein characteristics but will require rethinking of the existing data sets at the nucleotide level.

5.2. Learning from Protein Dynamics. Protein dynamics profoundly influence biological phenomena and properties ranging from enzyme mechanisms to protein stability and must therefore be taken into account when designing enzymes.[275] The value of analyzing protein ensembles rather than single-protein structures has been demonstrated in contexts including predicting thermal stability[276] and identifying the long-range conformational dynamic effect on residues involved in substrate reorientation[277] and reactivity-promoting regions.[278] Therefore, tools that can generate dynamic data without extensive computational or experimental data collection would be extremely valuable. This has motivated the development of new methods for generating representative ensembles from structure[279] or sequence data.[280]

Despite the growing availability of protein dynamics data from sources such as MD simulations, current methods for predicting protein properties are mainly based on single sequences or static structures. This is partly because several challenges must be overcome when MD results are incorporated into the training and inference phases of classical ML pipelines. The first challenge concerns data representation: predictive models typically work with static structures. One possible strategy could be to include a limited set of quantities derived from a trajectory, *e.g.*, the flexibility of structural elements expressed in terms of their B-factors, in the set of input features. Another possibility would be to select a subset of conformations to be used during training. For example, one might use apo and holo structures or representative conformations chosen with various clustering methods or Markov State Modeling. However, the applicability of these methods to training data sets of multiple proteins has yet to be explored.

An interesting recent approach that could help with data processing is "dataset distillation",[281] which builds on research related to distillation of neural network models.[282] In data set distillation, the goal is to distill a larger data set into a smaller one of potentially artificial examples so that models trained on the smaller data set can match the performance of those trained on the larger one. This process was first demonstrated in the image recognition domain by distilling the well-known MNIST data set[283] of 60k images into just 10 synthetic images (one per class); models trained on the distilled data set

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

achieved performance closely approaching that of models trained on the original data set.[281] At present, data set distillation has been most beneficial for models that are retrained multiple times, such as those used in **continual learning** or NN architecture searching. The use of data set distillation is conditioned on the existence of large data sets and relevant models that can be trained on those data. So far, relatively little work has been done on distilling temporal data such as videos,[284] but as soon as data set distillation demonstrates its utility in the domain of video analysis, we expect this technology to become applicable to processing of MD trajectories as well. Data set distillation techniques have recently been comprehensively reviewed by Yu et al.[285] and Lei et al.[286]

The lack of training data and benchmarks appears to be another fundamental barrier to the systematic integration of MD data into ML-based protein engineering pipelines. Compiling data sets of MD trajectories presents both technical and scientific challenges resulting from the inconsistent file formats due to the use of different simulation packages,[287] huge file sizes, and the fact that MD data are rarely published. Furthermore, MD trajectories can be very sensitive to the choice of force fields[288] and other settings. This high variability of simulation data makes compiling large, consistent, high-quality data sets truly challenging. The application of the FAIR principles[289] to the publication of biomolecular simulation results is thus an important first step toward building relevant data sets and benchmarks and will encourage further development of the field by greatly increasing data availability. The initiative of publishing the MD trajectories is also called upon by Tiemann et al., who performed an extensive MD data mining exercise and demonstrated the utility of publicly accessible data.[290] The prioritization of building data sets for specific proteins or protein families appears to be a reasonable next step, followed by exploring the possibility of transferring knowledge between different protein families.[291]

5.3. Missing Gold-Standard Data Sets. High-quality data benchmarks are major drivers of progress in ML research.[292] For instance, the remarkable progress in image classification can largely be attributed to the existence of well-prepared and maintained benchmarks: an early example was the MNIST collection of hand-written digits,[283,293] which was later complemented by the PASCAL visual object classification (VOC) challenge data set of real photographs[293] and the ImageNet data set.[294] MNIST enabled the first successes of deep learning on hand-written digits,[295] while the PASCAL VOC data set and the associated challenge established standard benchmarking practices in computer vision research, including the use of a hidden test set. Over the past decade, the benchmarking of progressively more advanced convolutional neural networks on ImageNet has produced models with superhuman classification performance.[296] Here it is important to stress the difference between a data set and a benchmark data set: to qualify as a benchmark, a data set must satisfy stringent quality criteria, have well-defined benchmarking tasks and performance metrics, and have a predefined split into training and test sets.[297] Moreover, the performance of all models on the MNIST data set is measured in terms of the percentage of wrong classifications. Such universally accepted benchmarking criteria enable clear and fair evaluations of the practical performance of proposed models, eliminate the need to spend time on data preparation, and introduce an element of competition into model development.

While such benchmarking practices are standard in traditional ML domains, they remain far from common in protein engineering for several reasons. First, the complexity of the domain presents challenges in collecting data and ensuring their quality. The quality of ImageNet was guaranteed by manual verification of each image, which was achieved by creating a special interactive web site to which any nonexpert could contribute. The bulk of the annotation of one million images was done using annotation services, such as Amazon Mechanical Turk. However, protein data require much more involved manual curation by domain experts, making such approaches unusable. Automated curation may be feasible for some types of data, *e.g.*, MMseqs2 and Foldseek enable fast deduplication and clustering of sequences[80,298] and monomeric structures.[80,298] However, it is extremely difficult for data sets focusing on the catalytic activity, specificity, enantioselectivity, or solubility. Large-scale data cleaning also remains challenging for protein−protein interfaces, which are highly repetitive due to the redundancy of the PDB.[31,299,300]

Next, the selection of a suitable ML evaluation metric and data split may be challenging because of the inherent interdisciplinarity of protein engineering. The establishment of benchmark components requires a deep understanding of the intricacies of biochemistry as well as expertise in ML. For example, mutational data sets can be split at the level of specific substitutions, mutation sites, specific proteins, or protein families while ensuring that no entry in the test set is repeated in the training data. However, even in a simple setting using protein sequences, splitting at the protein level becomes challenging if the data set contains many homologs or protein variants, and many authors use different sequence identity cutoffs for clustering data before splitting. Protein structure-based learning introduces additional complexity into the process of defining splits. Distinct sequences may have very similar tertiary structures, necessitating the use of more advanced geometric or graph-based splitting conditions. However, such methods are rarely applied, leading to the use of a wide variety of splits that limits model comparability.[156] For example, a widely used data set that was reported to suffer from improper splitting[301] is PDBBind.[302] Simple random splits may result in data leakage, especially when using highly redundant or nonuniform data sets.[77,303,304]

The Critical Assessment of Structure Prediction (CASP) is the best example of a well-established benchmark in protein science.[305] It has been produced and maintained by a community-driven effort that has been ongoing for almost three decades, enabled the success of AlphaFold2, and is continually being extended to new and more complex tasks.[99] Despite the existence of several other CASP-inspired benchmarks such as Critical Assessment of Prediction of Interactions (CAPRI) and Critical Assessment of Function Annotation (CAFA),[306,307] standard benchmarks for protein design are still lacking. This problem is attracting attention; for example, Dallago et al. recently introduced the Fitness Landscape Inference for Proteins (FLIP) benchmark,[303] which addresses three protein engineering problems by including: (i) an almost complete mutational landscape (149 361 of 160 000 variants) of four strongly epistatic residues of the G protein, (ii) the more diverse and sparsely sampled landscape of the AAV capsid protein, and (iii) the highly diverse thermostability landscape of proteins from different domains of life. Similar benchmarks are also appearing in related disciplines such as genomics[308] and drug discovery.[75,309] Calls for defining clear

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

criteria for the study of epistasis and residue coevolution patterns can be found in the literature;[175] these criteria could enable the emergence of new benchmarks. We hope to see more initiatives creating and supporting benchmarking in the future, as they could dramatically accelerate progress in ML for protein design.

5.4. Poor Transfer of Knowledge from Concept to Application. Even when authors have performed appropriate independent method validation and convinced readers of the superiority of their tool, the transfer of knowledge from published methodologies to new applications seems to be frustratingly slow. Despite the rapidly growing number of publications describing applications of ML in protein engineering, surprisingly few studies have followed up on published methods and applied them to new protein targets. The main obstacle to such work appears to be the limited accessibility of ML methods to researchers without expertise in computer science, *e.g.*, biochemists. One established scientific publishing standard for new ML methods is that protocols and scripts should be included with the submitted manuscript.[310] While this improves the peer-review process, potential future users of such tools are generally less comfortable running such scripts than reviewers. Therefore, if a new method is available only as a collection of scripts in a GitHub repository, it is unlikely to have much impact outside the narrow community of ML developers.

Creating at least a minimalistic user-friendly interface is thus vital for making methods widely accessible to users other than their developers, even though it generally requires additional work from the developers that is outside the scope of their research objectives. The explosive growth in the number of users of ChatGPT 3.5 and later versions has been partially attributed to its simple dialogue-like interface, which is accessible even to users without knowledge of ML. Similarly, the AlphaFold2 release was followed by the dissemination of a Google Colab notebook,[311] allowing it to be used by researchers in a way that is much more user-friendly than downloading code from GitHub. In Loschmidt Laboratories, we have been developing user-friendly tools for over two decades (https://loschmidt.chemi.muni.cz/portal/), and we often hear how crucial the ease of use of these tools is to our users. We ensure this ease of use by making efforts to support popular input formats (ideally, using file types that a typical user will have at hand), creating intuitive and well-guided settings (ideally, with a default setup that will provide reasonable results in most use cases), and providing comprehensive and easily understood reporting of results. We also strongly encourage other developers to invest time into making their tools accessible to other communities, such as enzymologists or biochemists. Moreover, it is usually helpful to contact past or potential users to understand how they see the method and what functionality they would benefit from in addition to the method itself. This could include fetching sequences and structures from databases, submitting multiple sequences or mutations as a single input, suggesting designs for experimental validation, or even allowing output graphics to be easily transferred to a publication.

We also encourage authors to be more open about the limitations of their tools in publications. When published case studies use additional steps, *e.g.*, manual fine-tuning, these should be discussed explicitly in the main text of a publication or in instructions for using tools. While there is pressure to present a tool's performance in the most favorable light, any cherrypicking will eventually disappoint future users and undermine the trust in the entire domain. To be successful, ML requires users and the wider community to trust its predictions and be convinced that the tools have been assessed fairly. Consequently, there are active efforts to improve reporting standards for ML tools in modern biology, and several clear guidelines have recently appeared to guide authors in preparing manuscripts dealing with ML.[12,310,312] However, there will be a lag phase before such rules are universally adopted.

6. FUTURE OPPORTUNITIES

6.1. Trustworthiness and Explainable AI. In many domains, including medicine and finance, it is vital for ML systems to be interpretable and explainable to build trust in the algorithms. Understanding the mechanisms behind making a prediction can also enable more rigorous verification of a tool or provide clues for follow-up decision-making. In this context, an explainable model is one that provides insights into the reasons for its predictions and decisions, for example, by showing which parts of the input have the greatest impact on predictions. Conversely, an interpretable model is one for which the internal process of making predictions can be readily understood by humans. This can be achieved by having an explicit decision pathway in a decision tree or easy-to-grasp feature weights in a simple linear equation. If ML algorithms are not mathematically complex, they are often intrinsically explicable and interpretable. However, interpreting and explaining their predictions becomes more challenging as the algorithms become more complicated. This is why models such as deep neural networks are often described as black boxes. The field of explainable artificial intelligence (XAI) seeks to overcome this challenge by creating explanations using analytics, saliency maps, or words to allow humans to understand why an ML algorithm makes a certain decision or prediction.

A range of XAI methods have been proposed in recent years[313,314] (Figure 4) and are discussed in more detail in two recent comprehensive reviews.[313,315] Two simple strategies for XAI are to use self-explainable white-box methods and to check how changing inputs affect the outputs of black-box networks. Feature importance methods are a notable class of white-box methods that achieve explainability by identifying essential features based on model parameters such as weights and coefficients. For example, in a linear regression algorithm, scientists can determine the importance of each feature based on the magnitude of the associated coefficients. Propagation-based approaches are another class of white-box methods that are often used for similar purposes in deep learning. One of the recent methods in this class is layer-wise relevance propagation, which propagates predictions from the output to the input using propagation rules evaluated at each node of a neural network.[316] As this method relies on simple formulas, it does not require computationally demanding sampling and is relatively robust to noise and other artifacts during training.[313]

For black-box models, explainability and interpretability are generally achieved by analyzing the input-output behavior. For example, Shapley values estimate the contribution of each feature by evaluating its marginal impact on the output. Unfortunately, this approach requires significant computational resources and scales poorly with increasing numbers of features.[317] Another strategy is to approximate a black-box model with a more interpretable analogue. This approach is

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

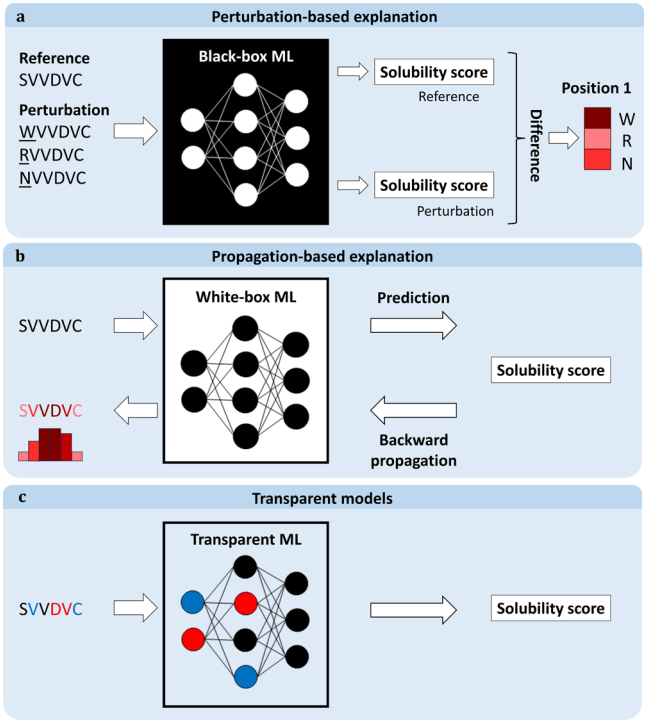


Figure 4. An overview of explainable AI methods. The figure illustrates the concepts using the task of predicting protein solubility from primary structure data as an example. (a) Perturbation methods check the effect of changes in input data on output; a significant change in output causes the corresponding input to be marked as crucial for prediction (darker red colors on the output indicate a higher difference in prediction and hence higher importance for the output). Perturbed amino acids in the input are underlined. (b) Propagation methods use the network structure and move from the prediction to the input to determine which parts of the input have the greatest impact on the prediction (here, darker colors indicate greater impact on the prediction). (c) Transparent networks are designed to be interpretable, for example, by specific choices of their architecture or the individual building blocks (*e.g.*, filters in a convolutional network might be interpreted as specific relevant sequence motifs identified during training). Here different parts of the network (shown in different colors) are related to different parts of the input sequence (illustrated in the same colors).

exemplified by the LIME method,[318] which mimics the predictions of any classifier in an interpretable and faithful manner by locally approximating the decision boundary using a linear regression or a decision tree. Another black-box method explains model predictions by perturbing the input data and analyzing the differences between the actual output and the output for the perturbed data set. If a particular perturbation of the input produces large differences in output, the changed parts of the data set are marked as being essential for prediction.[319]

Another approach to incorporating explainability into ML pipeline is to pursue explainability during the development of the pipeline rather than when the predictor is released, *e.g.*, by using transparent neural networks.[320] Such neural networks are designed to be more interpretable and understandable by humans. Prior knowledge, such as existing biological knowledge and experimental data, can help scientists develop such models. For example, a convolutional neural network could be designed to learn filters that correspond to known protein structural motifs.[321] Alternatively, one could use knowledge-primed neural networks in which nodes correspond to proteins that are connected based on prior knowledge.[322]

The ML methods that are increasingly being applied in protein engineering are often complex and challenging to interpret. Because wet-lab validation of ML-designed proteins and enzymes is expensive and time-consuming, it is vital to reinforce the credibility of AI-assisted protein engineering and ensure that experimentalists can be confident that the methods will produce designs with a good chance of success in the lab. XAI can strengthen credibility by providing insights into the decision-making processes of ML models in protein engineering.[323] It can also help scientists to improve their algorithms by revealing mistakes and biases.[324] The use of XAI to explain deep learning networks has therefore recently attracted interest in areas of drug discovery, chemistry, and protein engineering including active ligand searching,[325] prediction of enzyme EC numbers,[326] and identifying residues that indicate transitions between active and inactive states in GPCR receptors.[327]

In protein engineering, explainable AI is primarily applied for the analysis of predictions from ML models with the aim of obtaining novel biochemistry knowledge. Namely, Tan et al. introduced ExplainableFold,[328] a concept designed to improve the understanding of deep learning-based protein structure prediction models such as AlphaFold by residue deletions and residue substitutions. Essentially, the core objective of ExplainableFold is to uncover influential residues responsible for maintaining or altering a folded protein structure. More generally, it was also proposed that the application of tools such as exBERT,[329] originally designed for visualizing internal representations of transformers, could be employed in protein-trained transformers to highlight relationships among amino acids.[330] Ultimately, the use of explainable AI for protein design is still in its early phase and we have yet to see its main applications in protein engineering pipelines.

6.2. Identification of Hidden Biases. ML models can be biased by the data used in their training, which can reduce their overall accuracy or cause their performance to vary substantially across the data input space, *e.g.*, the protein sequence space or the space of protein structures. In supervised learning, special attention must be paid to the problem of class imbalance,[331−333] which occurs when there are more training examples for some classes than for others. In such cases, the model might overpredict the major classes.[334] In unsupervised learning, these biases might be harder to spot and quantify. For example, in natural language processing, large self-supervised language models are typically trained on vast amounts of text that are often sourced from the Internet without thorough data curation. If left unchecked, these models can reportedly generate unjust or oppressive language, which promotes discrimination, exclusion, and toxicity.[335] The risk of compromising model performance by using biased training data is as relevant in enzyme engineering as in any other field.[31] However, in contrast to natural languages, it is not directly apparent what adverse biases could be adopted by models in enzymology, as there is no quick and simple strategy to directly validate outputs. Therefore, responsibility and caution are advised, especially when different sources of data are combined, *e.g.*, with the data from human genomic databases, which were reported to be racially biased.[336]

A typical example of a bias in enzymatic data arises in the task of predicting mutational stability, where a predictor given an original and mutated sequence as inputs should report a single number corresponding to the predicted ΔΔG. In

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

general, a random mutation of a natural protein is more likely to be destabilizing than stabilizing,[337] and this bias often propagates to stability data sets, leading to an issue similar to the class imbalance problem resulting in the overprediction of destabilizing mutations. To combat such effects, it has become standard to exploit the antisymmetry of mutational stability, which arises from the physical principle described by the following equation:[338] $\Delta\Delta G(\mathrm{WT} \rightarrow \mathrm{Mut}) = -\Delta\Delta G(\mathrm{Mut} \rightarrow \mathrm{WT})$. That is, the change in ΔG upon mutating a residue is equal to the negative change in ΔG that would be induced by the hypothetical inverse mutation. This property makes it possible to augment the training data set by adding artificial inverse mutations from the mutant sequence (Mut) to the wild-type sequence (WT) to obtain a balanced data set. Antisymmetry can also be incorporated by design, ensuring that the property is enforced even when using an imbalanced training set.[339] Interestingly, efforts to solve the problem of predicting protein stability changes upon mutation have driven progress in other sensitivity studies, *e.g.*, the use of reduced amino acid alphabets to account for bias in the representation of mutations in the training data,[131] structural sensitivity,[340] or the "scaffold bias" of using crystallographic structures instead of AlphaFold models in ensembles.[276] Another task where data set biases were reported is structure-based virtual screening. This problem was recently tackled using an ML-based scoring function in which the authors took care to ensure that feature importance was consistent with human knowledge; this forced the model to learn relevant features regardless of the present biases, leading to better generalizability.[341]

In many cases, however, even a detailed understanding of the studied task will not reveal straightforward paths for uncovering and fixing biases because problems often arise from intricate interdependencies between data points that researchers are unaware of. To combat such issues, the concept of multicalibration has been proposed.[342] The aim of multicalibration is to make predictors perform more uniformly across different data subclasses. However, the complexity of the proposed approach is linear with respect to the number of possible subsets of the training data and, thus, exponential with respect to the number of training samples. To alleviate this high computational cost, other studies proposed low-degree multicalibration by drawing on the multicalibration and multiaccuracy approaches.[343,344] These concepts are currently mainly being studied within the domain of algorithmic fairness,[345] which is largely concerned with ML-based predictors that process personal data on human individuals. It is interesting to see if these concepts will be useful in the context of enzymology, for example, by ensuring that learned predictors have similar performance for different protein families.

6.3. Other Promising Methods. One current trend in machine learning is that models are increasing in size in parallel with the increasing amount of data on which they are trained. Models such as ChatGPT and DALL-E[38] are shining examples of this trend. To increase the size of the data sets available for model training in protein engineering, it may be necessary to use as much available experimental data as possible, accepting that experimental data sets will inevitably vary widely in quality and size. To bridge the gap between small high-quality and large low-quality data sets, meta-learning appears to be a promising strategy. In such an approach, nested optimization could be employed for training on a large set of noisy examples in the inner loop and a small set of trusted examples in the outer loop, thus suppressing the impact of the noise in the larger data set.[346]

Molecular simulations using MD and quantum mechanical methods could be another valuable source of data for training large, data-hungry models. However, these simulations are very computationally demanding, as demonstrated by Musaelian et al., who recently used machine-learned potentials in simulations of biomolecules using 5120 A100 GPUs in parallel.[347] The hardware requirements of MD using such large data sets could be met using purpose-built supercomputers such as Anton 3.[348] Advances in quantum computing (QC) technology could also make large-scale molecular simulations more accessible, as well as benefit generative ML tasks[349] and improve the generalizability of models trained on limited data.[350] However, the greatest benefit of QC in enzyme engineering will likely result from its expected ability to greatly accelerate quantum and molecular mechanics simulations.[351,352]

Given the trend toward ever-larger models, it is important to consider the cost of their training and the CO_2 emissions resulting from the training and inference process.[353] This issue can be addressed by adopting energy-saving ML practices,[354] implementing architectural redundancy to bypass lengthy training processes,[355] or obviating the need for large model training by narrowing problems down to simple linear combinations of system-relevant physical properties.[356] Another important recent trend in machine learning involves lightweight adaptation of large pretrained models to new tasks using adaptor layers.[201,357] These methods change only a small fraction of the parameters of the large scale pretrained model for the new task, leaving the vast majority untouched. This allows large models to be adapted to new tasks at a fraction of their initial training costs; the time required for adaptation may be as little as several hours on a single eight-GPU machine,[358,359] eliminating the need for the supercomputer infrastructure used to train the original model. This significantly accelerates model adaptation and makes such training and related research accessible to a wide community of researchers who may lack supercomputer access.

Another promising area of application for ML is further automation of enzyme engineering pipelines, for example, by using **reinforcement learning** (RL) to select structural or biophysical motifs that are important for a target property.[360] Several recent advances in RL have focused on small molecule design.[361] However, when combined with recent advances in protein science, RL has also shown promise in the design of protein complexes[362] or peptide binders with high affinities.[363] Based on this expansion of its applicability domains, we believe that RL could be similarly useful in enzyme engineering. Enabling more user-friendly pipeline interaction, for example, by creating ChatGPT-like interfaces to simplify the control and running of experiments, can potentially further contribute to the automation of the workflows.

7. CONCLUSIONS

We live in exciting times that provide many opportunities to explore new horizons in enzyme engineering and machine learning. Many ambitious tasks are already being tackled by cutting-edge data-driven algorithms and approaches, often inspired by their spectacular performance in other domains. The tools created on their basis are already playing integral roles in studies aiming to discover and improve biocatalysts. On the other hand, many more goals are still waiting for the

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895

appearance of suitable data sets and data processing protocols to be eventually leveraged by machine learning. There are many challenges ahead, including the creation of reliable and user-friendly tools for generating promising protein designs by users with limited ML knowledge, addressing multiple tasks in one pipeline, incorporating protein dynamics in ML pipelines, understanding the effects of insertions, deletions, or unnatural amino acids, increasing the interpretability of the models, and revealing hidden data biases. One of the most significant takeaways from the success of machine learning in other domains, such as natural language processing or computer vision, is the role of large-scale data collection and curation. As biochemistry data tend to be more challenging to acquire than, for example, images or text, developing mechanisms for open sharing and aggregation of data sets across the entire scientific community could have game-changing effects. Building and maintaining community-wide training data sets and benchmarks, including such ambitious data types as protein dynamics, could create entirely new ways for designing proteins that would shape the future of biotechnology.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acscatal.3c02743.

Selected recent case studies combining machine learning and protein engineering and a list of databases and data sets for machine learning applications in protein engineering (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Tomas Pluskal − *Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences, 160 00 Prague, Czech Republic;* orcid.org/0000-0002-6940-3006; Email: tomas.pluskal@uochb.cas.cz

Josef Sivic − *Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 160 00 Prague, Czech Republic*; Email: Josef.Sivic@cvut.cz

Stanislav Mazurenko − *Loschmidt Laboratories, Department of Experimental Biology and RECETOX, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic; International Clinical Research Center, St. Anne's University Hospital Brno, 656 91 Brno, Czech Republic;* orcid.org/0000-0003-3659-4819; Email: mazurenko@mail.muni.cz

Authors

Petr Kouba − *Loschmidt Laboratories, Department of Experimental Biology and RECETOX, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic; Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 160 00 Prague, Czech Republic; Faculty of Electrical Engineering, Czech Technical University in Prague, 166 27 Prague, Czech Republic;* orcid.org/0000-0002-9979-4159

Pavel Kohout − *Loschmidt Laboratories, Department of Experimental Biology and RECETOX, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic; International Clinical Research Center, St. Anne's University Hospital Brno, 656 91 Brno, Czech Republic*

Faraneh Haddadi − *Loschmidt Laboratories, Department of Experimental Biology and RECETOX, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic; International Clinical Research Center, St. Anne's University Hospital Brno, 656 91 Brno, Czech Republic*

Anton Bushuiev − *Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 160 00 Prague, Czech Republic;* orcid.org/0009-0007-4783-6584

Raman Samusevich − *Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 160 00 Prague, Czech Republic; Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences, 160 00 Prague, Czech Republic*

Jiri Sedlar − *Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 160 00 Prague, Czech Republic*

Jiri Damborsky − *Loschmidt Laboratories, Department of Experimental Biology and RECETOX, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic; International Clinical Research Center, St. Anne's University Hospital Brno, 656 91 Brno, Czech Republic;* orcid.org/0000-0002-7848-8216

Complete contact information is available at: https://pubs.acs.org/10.1021/acscatal.3c02743

Author Contributions

#These authors contributed equally to this study.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

T.P. is supported by the Czech Science Foundation (GA CR) Grant 21-11563M and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement 891397. P. Kouba and P. Kohout are holders of the Brno Ph.D. Talent scholarship funded by the Brno City Municipality and the JCMM. P. Kouba, A.B., R.S., J. Sedlar, and J. Sivic were supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468). This work was also supported by Czech Ministry of Education, Youth and Sports [CETOCOEN Excellence CZ.02.1.01/0.0/0.0/17_043/0009632, ESFRI RECETOX RI LM2023069, ESFRI ELIXIR LM2023055]; Technology Agency of the Czech Republic under the NCC Programme from the state budget (RETEMED TN02000122) and under the NRP from the EU RRF (TEREP TN02000122/001N); CETOCOEN EXCELLENCE Teaming project supported by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 857560; the project National Institute for Cancer Research (Programme EXCELES, no. LX22NPO5102) (funded by the European Union) Next Generation EU; and COST (European Cooperation in Science and Technology) [COST Action COZYME CA21162]. This publication reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information it contains.

■ REFERENCES

(1) Wu, S.; Snajdrova, R.; Moore, J. C.; Baldenius, K.; Bornscheuer, U. T. Biocatalysis: Enzymatic Synthesis for Industrial Applications. *Angew. Chem., Int. Ed. Engl.* **2021**, *60* (1), 88−119.

(2) Bell, E. L.; Finnigan, W.; France, S. P.; Green, A. P.; Hayes, M. A.; Hepworth, L. J.; Lovelock, S. L.; Niikura, H.; Osuna, S.; Romero, E.; Ryan, K. S.; Turner, N. J.; Flitsch, S. L. Biocatalysis. *Nat. Rev. Methods Primers* **2021**, *1*, 46.
(3) Silvestre, B. S.; Ţîrcă, D. M. Innovations for Sustainable Development: Moving toward a Sustainable Future. *J. Clean. Prod.* **2019**, *208*, 325−332.
(4) Tiso, T.; Winter, B.; Wei, R.; Hee, J.; de Witt, J.; Wierckx, N.; Quicker, P.; Bornscheuer, U. T.; Bardow, A.; Nogales, J.; Blank, L. M. The Metabolic Potential of Plastics as Biotechnological Carbon Sources - Review and Targets for the Future. *Metab. Eng.* **2022**, *71*, 77−98.
(5) Pimviriyakul, P.; Wongnate, T.; Tinikul, R.; Chaiyen, P. Microbial Degradation of Halogenated Aromatics: Molecular Mechanisms and Enzymatic Reactions. *Microb. Biotechnol.* **2020**, *13* (1), 67−86.
(6) Marques, S. M.; Planas-Iglesias, J.; Damborsky, J. Web-Based Tools for Computational Enzyme Design. *Curr. Opin. Struct. Biol.* **2021**, *69*, 19−34.
(7) Chang, C.; Deringer, V. L.; Katti, K. S.; Van Speybroeck, V.; Wolverton, C. M. Simulations in the Era of Exascale Computing. *Nat Rev Mater* **2023**, *8* (5), 309−313.
(8) Pyzer-Knapp, E. O.; Pitera, J. W.; Staar, P. W. J.; Takeda, S.; Laino, T.; Sanders, D. P.; Sexton, J.; Smith, J. R.; Curioni, A. Accelerating Materials Discovery Using Artificial Intelligence, High Performance Computing and Robotics. *npj Comput. Mater.* **2022**, *8*, 84.
(9) Singh, V.; Patra, S.; Murugan, N. A.; Toncu, D.-C.; Tiwari, A. Recent Trends in Computational Tools and Data-Driven Modeling for Advanced Materials. *Mater. Adv.* **2022**, *3* (10), 4069−4087.
(10) Greener, J. G.; Kandathil, S. M.; Moffat, L.; Jones, D. T. A Guide to Machine Learning for Biologists. *Nat. Rev. Mol. Cell Biol.* **2022**, *23* (1), 40−55.
(11) Beller, M.; Bender, M.; Bornscheuer, U. T.; Schunk, S. Catalysis − Far from Being a Mature Technology. *Chem. Ing. Tech.* **2022**, *94* (11), 1559−1559.
(12) Oza, V. H.; Whitlock, J. H.; Wilk, E. J.; Uno-Antonison, A.; Wilk, B.; Gajapathy, M.; Howton, T. C.; Trull, A.; Ianov, L.; Worthey, E. A.; Lasseigne, B. N. Ten Simple Rules for Using Public Biological Data for Your Research. *PLoS Comput. Biol.* **2023**, *19* (1), No. e1010749.
(13) Mazurenko, S.; Prokop, Z.; Damborsky, J. Machine Learning in Enzyme Engineering. *ACS Catal.* **2020**, *10* (2), 1210−1223.
(14) Strokach, A.; Kim, P. M. Deep Generative Modeling for Protein Design. *Curr. Opin. Struct. Biol.* **2022**, *72*, 226−236.
(15) Ding, W.; Nakai, K.; Gong, H. Protein Design via Deep Learning. *Brief. Bioinform.* **2022**, *23* (3), No. bbac102, DOI: 10.1093/bib/bbac102.
(16) Pan, X.; Kortemme, T. Recent Advances in de Novo Protein Design: Principles, Methods, and Applications. *J. Biol. Chem.* **2021**, *296*, No. 100558.
(17) Chandra, A.; Tünnermann, L.; Löfstedt, T.; Gratz, R. Transformer-Based Deep Learning for Predicting Protein Properties in the Life Sciences. *Elife* **2023**, *12*, No. e82819, DOI: 10.7554/eLife.82819.
(18) Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A Survey of Transformers. *AI Open* **2022**, *3*, 111−132.
(19) Zhang, X.-M.; Liang, L.; Liu, L.; Tang, M.-J. Graph Neural Networks and Their Current Applications in Bioinformatics. *Front. Genet.* **2021**, *12*, No. 690049.
(20) Bronstein, M. M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv [cs.LG]* **2021**, DOI: 10.48550/arXiv.2104.13478.
(21) Alzubaidi, L.; Zhang, J.; Humaidi, A. J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M. A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J Big Data* **2021**, *8*, 53.
(22) Calin, O. *Deep Learning Architectures*; Springer, 2020.
(23) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.
(24) Bordin, N.; Dallago, C.; Heinzinger, M.; Kim, S.; Littmann, M.; Rauer, C.; Steinegger, M.; Rost, B.; Orengo, C. Novel Machine Learning Approaches Revolutionize Protein Knowledge. *Trends Biochem. Sci.* **2023**, *48* (4), 345−359.
(25) Mowbray, M.; Savage, T.; Wu, C.; Song, Z.; Cho, B. A.; Del Rio-Chanona, E. A.; Zhang, D. Machine Learning for Biochemical Engineering: A Review. *Biochemical Eng. J.* **2021**, *172*, 108054.
(26) Hon, J.; Marusiak, M.; Martinek, T.; Kunka, A.; Zendulka, J.; Bednar, D.; Damborsky, J. SoluProt: Prediction of Soluble Protein Expression in Escherichia Coli. *Bioinformatics* **2021**, *37* (1), 23−28.
(27) Wu, K. E.; Yang, K. K.; van den Berg, R.; Zou, J. Y.; Lu, A. X.; Amini, A. P. Protein Structure Generation via Folding Diffusion. *arXiv [q-bio.BM]* **2022**, DOI: 10.48550/arXiv.2209.15611.
(28) Watson, J. L.; Juergens, D.; Bennett, N. R.; Trippe, B. L.; Yim, J.; Eisenach, H. E.; Ahern, W.; Borst, A. J.; Ragotte, R. J.; Milles, L. F.; Wicky, B. I. M.; Hanikel, N.; Pellock, S. J.; Courbet, A.; Sheffler, W.; Wang, J.; Venkatesh, P.; Sappington, I.; Torres, S. V.; Lauko, A.; De Bortoli, V.; Mathieu, E.; Ovchinnikov, S.; Barzilay, R.; Jaakkola, T. S.; DiMaio, F.; Baek, M.; Baker, D. De Novo Design of Protein Structure and Function with RFdiffusion. *Nature* **2023**, *620*, 1089.
(29) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. *arXiv [q-bio.BM]* **2022**, DOI: 10.48550/arXiv.2210.01776.
(30) Guo, Z.; Liu, J.; Wang, Y.; Chen, M.; Wang, D.; Xu, D.; Cheng, J. Diffusion Models in Bioinformatics: A New Wave of Deep Learning Revolution in Action. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2302.10907.
(31) Shroff, R.; Cole, A. W.; Diaz, D. J.; Morrow, B. R.; Donnell, I.; Annapareddy, A.; Gollihar, J.; Ellington, A. D.; Thyer, R. Discovery of Novel Gain-of-Function Mutations Guided by Structure-Based Deep Learning. *ACS Synth. Biol.* **2020**, *9* (11), 2927−2935.
(32) Zhang, Z.; Xu, M.; Chenthamarakshan, V.; Lozano, A.; Das, P.; Tang, J. Enhancing Protein Language Models with Structure-Based Encoder and Pre-Training. *arXiv (Quantitative Biology.Quantitative Methods)*, March 11, 2023, 2303.06275, ver. 1. DOI: 10.48550/arXiv.2303.06275
(33) Diaz, D. J.; Gong, C.; Ouyang-Zhang, J.; Loy, J. M.; Wells, J.; Yang, D.; Ellington, A. D.; Dimakis, A.; Klivans, A. R. Stability Oracle: A Structure-Based Graph-Transformer for Identifying Stabilizing Mutations. *bioRxiv (Biochemistry)*, March 22, 2023, 2023.05.15.540857. DOI: 10.1101/2023.05.15.540857.
(34) Ferruz, N.; Heinzinger, M.; Akdel, M.; Goncearenco, A.; Naef, L.; Dallago, C. From Sequence to Function through Structure: Deep Learning for Protein Design. *Comput. Struct. Biotechnol. J.* **2023**, *21*, 238−250.
(35) Madani, A.; Krause, B.; Greene, E. R.; Subramanian, S.; Mohr, B. P.; Holton, J. M.; Olmos, J. L., Jr; Xiong, C.; Sun, Z. Z.; Socher, R.; Fraser, J. S.; Naik, N. Large Language Models Generate Functional Protein Sequences across Diverse Families. *Nat. Biotechnol.* **2023**, *41*, 1099.
(36) Li, Y.; Rezaei, M. A.; Li, C.; Li, X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. In *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, November 18−21, 2019; IEEE, 2019; pp 303−310.
(37) Park, S.; Seok, C. GalaxyWater-CNN: Prediction of Water Positions on the Protein Structure by a 3D-Convolutional Neural Network. *J. Chem. Inf. Model.* **2022**, *62* (13), 3157−3168.
(38) Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv [cs.CV]* **2022**, DOI: 10.48550/ARXIV.2204.06125.
(39) Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv [cs.CV]* **2021**, DOI: 10.48550/arXiv.2112.10752.
(40) Schneuing, A.; Du, Y.; Harris, C.; Jamasb, A.; Igashov, I.; Du, W.; Blundell, T.; Lió, P.; Gomes, C.; Welling, M.; Bronstein, M.;

Correia, B. Structure-Based Drug Design with Equivariant Diffusion Models. *arXiv [q-bio.BM]* **2022**, DOI: 10.48550/arXiv.2210.13695.
(41) Igashov, I.; Stärk, H.; Vignac, C.; Satorras, V. G.; Frossard, P.; Welling, M.; Bronstein, M. M.; Correia, B. Equivariant 3D-Conditional Diffusion Models for Molecular Linker Design. *OpenReview*, February 1, 2023. https://openreview.net/forum?id=cnsHSSLnHVV.
(42) Yang, A.; Nagrani, A.; Seo, P. H.; Miech, A.; Pont-Tuset, J.; Laptev, I.; Sivic, J.; Schmid, C. Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, June 2022, 2023; Computer Vision Foundation, 2023; pp 10714−10726.
(43) Huang, C.; Wu, Z.; Wen, J.; Xu, Y.; Jiang, Q.; Wang, Y. Abnormal Event Detection Using Deep Contrastive Learning for Intelligent Video Surveillance System. *IEEE Trans. Ind. Inf.* **2022**, *18* (8), 5171−5179.
(44) Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; Salimans, T. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv [cs.CV]* **2022**, DOI: 10.48550/arXiv.2210.02303.
(45) Villegas, R.; Babaeizadeh, M.; Kindermans, P.-J.; Moraldo, H.; Zhang, H.; Saffar, M. T.; Castro, S.; Kunze, J.; Erhan, D. Phenaki: Variable Length Video Generation from Open Domain Textual Descriptions. *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, May 1−5, 2023; OpenReview, 2023. https://openreview.net/pdf?id=vOEXS39nOF
(46) Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; Parikh, D.; Gupta, S.; Taigman, Y. Make-A-Video: Text-to-Video Generation without Text-Video Data. *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, May 1−5, 2023; OpenReview, 2023. https://openreview.net/pdf?id=nJfylDvgzlq
(47) Hung, M.; Lauren, E.; Hon, E. S.; Birmingham, W. C.; Xu, J.; Su, S.; Hon, S. D.; Park, J.; Dang, P.; Lipsky, M. S. Social Network Analysis of COVID-19 Sentiments: Application of Artificial Intelligence. *J. Med. Internet Res.* **2020**, *22* (8), No. e22590.
(48) Bryant, P.; Pozzati, G.; Elofsson, A. Improved Prediction of Protein-Protein Interactions Using AlphaFold2. *Nat. Commun.* **2022**, *13*, 1265.
(49) Muzio, G.; O'Bray, L.; Borgwardt, K. Biological Network Analysis with Deep Learning. *Brief. Bioinform.* **2021**, *22* (2), 1515−1530.
(50) Chen, J.; Zheng, S.; Zhao, H.; Yang, Y. Structure-Aware Protein Solubility Prediction from Sequence through Graph Convolutional Network and Predicted Contact Map. *J. Cheminform.* **2021**, *13*, 7.
(51) Jiang, J.; Wang, R.; Wei, G.-W. GGL-Tox: Geometric Graph Learning for Toxicity Prediction. *J. Chem. Inf. Model.* **2021**, *61* (4), 1691−1700.
(52) Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J.; Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; Lin, H. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22118−22133.
(53) Kawashima, S.; Pokarowski, P.; Pokarowska, M.; Kolinski, A.; Katayama, T.; Kanehisa, M. AAindex: Amino Acid Index Database, Progress Report 2008. *Nucleic Acids Res.* **2007**, *36*, D202−D205.
(54) ElAbd, H.; Bromberg, Y.; Hoarfrost, A.; Lenz, T.; Franke, A.; Wendorff, M. Amino Acid Encoding for Deep Learning Applications. *BMC Bioinformatics* **2020**, *21*, 235.
(55) Raimondi, D.; Orlando, G.; Vranken, W. F.; Moreau, Y. Exploring the Limitations of Biophysical Propensity Scales Coupled with Machine Learning for Protein Sequence Analysis. *Sci. Rep.* **2019**, *9*, 16932.
(56) Kandathil, S. M.; Greener, J. G.; Lau, A. M.; Jones, D. T. Ultrafast End-to-End Protein Structure Prediction Enables High-Throughput Exploration of Uncharacterized Proteins. *Proc. Natl. Acad. Sci. U. S. A.* **2022**, *119* (4), No. e2113348119, DOI: 10.1073/pnas.2113348119.
(57) Fasoulis, R.; Paliouras, G.; Kavraki, L. E. Graph Representation Learning for Structural Proteomics. *Emerg Top Life Sci* **2021**, *5* (6), 789−802.
(58) Hermosilla, P.; Schäfer, M.; Lang, M.; Fackelmann, G.; Vázquez, P. P.; Kozlíková, B.; Krone, M.; Ritschel, T.; Ropinski, T. Intrinsic-Extrinsic Convolution and Pooling for Learning on 3D Protein Structures. *Ninth International Conference on Learning Representations*, May 3−7, 2021; OpenReview, 2021.
(59) Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B. E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. *Nat. Commun.* **2022**, *13*, 2453.
(60) Gligorijević, V.; Renfrew, P. D.; Kosciolek, T.; Leman, J. K.; Berenberg, D.; Vatanen, T.; Chandler, C.; Taylor, B. C.; Fisk, I. M.; Vlamakis, H.; Xavier, R. J.; Knight, R.; Cho, K.; Bonneau, R. Structure-Based Protein Function Prediction Using Graph Convolutional Networks. *Nat. Commun.* **2021**, *12*, 3168.
(61) Gao, Z.; Jiang, C.; Zhang, J.; Jiang, X.; Li, L.; Zhao, P.; Yang, H.; Huang, Y.; Li, J. Hierarchical Graph Learning for Protein-Protein Interaction. *Nat. Commun.* **2023**, *14*, 1093.
(62) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5999.
(63) Fuchs, F.; Worrall, D.; Fischer, V.; Welling, M.; Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; Lin, H. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1970−1981.
(64) Detlefsen, N. S.; Hauberg, S.; Boomsma, W. Learning Meaningful Representations of Protein Sequences. *Nat. Commun.* **2022**, *13*, 1914.
(65) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences. *Proc. Natl. Acad. Sci. U. S. A.* **2021**, *118* (15), No. e2016239118, DOI: 10.1073/pnas.2016239118.
(66) Meier, J.; Rao, R.; Verkuil, R.; Liu, J.; Sercu, T.; Rives, A.; Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P. S.; Vaughan, J. W. Language Models Enable Zero-Shot Prediction of the Effects of Mutations on Protein Function. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29287−29303.
(67) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; Dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-Scale Prediction of Atomic-Level Protein Structure with a Language Model. *Science* **2023**, *379* (6637), 1123−1130.
(68) Zhang, Z.; Xu, M.; Jamasb, A.; Chenthamarakshan, V.; Lozano, A.; Das, P.; Tang, J. Protein Representation Learning by Geometric Structure Pretraining. *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, May 1−5, 2023; OpenReview, 2023. https://openreview.net/pdf?id=to3qCB3tOh9
(69) Fowler, D. M.; Fields, S. Deep Mutational Scanning: A New Style of Protein Science. *Nat. Methods* **2014**, *11* (8), 801−807.
(70) Vanella, R.; Kovacevic, G.; Doffini, V.; Fernández de Santaella, J.; Nash, M. A. High-Throughput Screening, next Generation Sequencing and Machine Learning: Advanced Methods in Enzyme Engineering. *Chem. Commun.* **2022**, *58* (15), 2455−2467.
(71) Morrison, K. L.; Weiss, G. A. Combinatorial Alanine-Scanning. *Curr. Opin. Chem. Biol.* **2001**, *5* (3), 302−307.
(72) Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language Models Are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877−1901.
(73) OpenAI. GPT-4 Technical Report. *arXiv (Computer Science.Computation and Language)*, March 27, 2023, 2303.08774. https://arxiv.org/abs/2303.08774.
(74) Luo, R.; Sun, L.; Xia, Y.; Qin, T.; Zhang, S.; Poon, H.; Liu, T.-Y. BioGPT: Generative Pre-Trained Transformer for Biomedical Text Generation and Mining. *Brief. Bioinform.* **2022**, *23* (6), No. bbac409, DOI: 10.1093/bib/bbac409.

(75) Zhu, Z.; Shi, C.; Zhang, Z.; Liu, S.; Xu, M.; Yuan, X.; Zhang, Y.; Chen, J.; Cai, H.; Lu, J.; Ma, C.; Liu, R.; Xhonneux, L.-P.; Qu, M.; Tang, J. TorchDrug: A Powerful and Flexible Machine Learning Platform for Drug Discovery. *arXiv [cs.LG]* **2022**, DOI: 10.48550/arXiv.2202.08320.
(76) Siedhoff, N. E.; Illig, A.-M.; Schwaneberg, U.; Davari, M. D. PyPEF-An Integrated Framework for Data-Driven Protein Engineering. *J. Chem. Inf. Model.* **2021**, *61* (7), 3463−3476.
(77) Draizen, E. J.; Murillo, L. F. R.; Readey, J.; Mura, C.; Bourne, P. E. Prop3D: A Flexible, Python-Based Platform for Machine Learning with Protein Structural Properties and Biophysical Data. *bioRxiv*, 2022, 2022.12.27.522071. DOI: 10.1101/2022.12.27.522071.
(78) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28* (1), 235−242.
(79) Chothia, C.; Lesk, A. M. The Relation between the Divergence of Sequence and Structure in Proteins. *EMBO J.* **1986**, *5* (4), 823−826.
(80) van Kempen, M.; Kim, S. S.; Tumescheit, C.; Mirdita, M.; Lee, J.; Gilchrist, C. L. M.; Söding, J.; Steinegger, M. Fast and Accurate Protein Structure Search with Foldseek. *Nat. Biotechnol.* **2023**, DOI: 10.1038/s41587-023-01773-0.
(81) Brookes, D.; Park, H.; Listgarten, J. Conditioning by Adaptive Sampling for Robust Design. In *Proceedings of the 36th International Conference on Machine Learning*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research, Vol. *97*; PMLR, 2019; pp 773−782.
(82) Sinai, S.; Wang, R.; Whatley, A.; Slocum, S.; Locane, E.; Kelsic, E. D. AdaLead: A Simple and Robust Adaptive Greedy Search Algorithm for Sequence Design. *arXiv (Computer Science.Machine Learning)*, October 5, 2020, 2010.02141, ver. 1. DOI: 10.48550/arXiv.2010.02141
(83) Ren, Z.; Li, J.; Ding, F.; Zhou, Y.; Ma, J.; Peng, J. Proximal Exploration for Model-Guided Protein Sequence Design. In *Proceedings of the 39th International Conference on Machine Learning*; Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S., Eds.; Proceedings of Machine Learning Research, Vol. *162*; PMLR, 2022; pp 18520−18536.
(84) Lipsh-Sokolik, R.; Khersonsky, O.; Schröder, S. P.; de Boer, C.; Hoch, S.-Y.; Davies, G. J.; Overkleeft, H. S.; Fleishman, S. J. Combinatorial Assembly and Design of Enzymes. *Science* **2023**, *379* (6628), 195−201.
(85) Yu, T.; Boob, A. G.; Volk, M. J.; Liu, X.; Cui, H.; Zhao, H. Machine Learning-Enabled Retrobiosynthesis of Molecules. *Nat. Catal.* **2023**, *6*, 137.
(86) Mistry, J.; Chuguransky, S.; Williams, L.; Qureshi, M.; Salazar, G. A.; Sonnhammer, E. L. L.; Tosatto, S. C. E.; Paladin, L.; Raj, S.; Richardson, L. J.; Finn, R. D.; Bateman, A. Pfam: The Protein Families Database in 2021. *Nucleic Acids Res.* **2021**, *49* (D1), D412−D419.
(87) Pandurangan, A. P.; Stahlhacke, J.; Oates, M. E.; Smithers, B.; Gough, J. The SUPERFAMILY 2.0 Database: A Significant Proteome Update and a New Webserver. *Nucleic Acids Res.* **2019**, *47* (D1), D490−D494.
(88) Sillitoe, I.; Bordin, N.; Dawson, N.; Waman, V. P.; Ashford, P.; Scholes, H. M.; Pang, C. S. M.; Woodridge, L.; Rauer, C.; Sen, N.; Abbasian, M.; Le Cornu, S.; Lam, S. D.; Berka, K.; Varekova, I. H.; Svobodova, R.; Lees, J.; Orengo, C. A. CATH: Increased Structural Coverage of Functional Space. *Nucleic Acids Res.* **2021**, *49* (D1), D266−D273.
(89) Alcántara, R.; Axelsen, K. B.; Morgat, A.; Belda, E.; Coudert, E.; Bridge, A.; Cao, H.; de Matos, P.; Ennis, M.; Turner, S.; Owen, G.; Bougueleret, L.; Xenarios, I.; Steinbeck, C. Rhea–a Manually Curated Resource of Biochemical Reactions. *Nucleic Acids Res.* **2012**, *40*, D754−D760.
(90) Schomburg, I.; Chang, A.; Schomburg, D. BRENDA, Enzyme Data and Metabolic Information. *Nucleic Acids Res.* **2002**, *30* (1), 47−49.
(91) Wittig, U.; Rey, M.; Weidemann, A.; Kania, R.; Müller, W. SABIO-RK: An Updated Resource for Manually Curated Biochemical Reaction Kinetics. *Nucleic Acids Res.* **2018**, *46* (D1), D656−D660.
(92) Wishart, D. S.; Li, C.; Marcu, A.; Badran, H.; Pon, A.; Budinski, Z.; Patron, J.; Lipton, D.; Cao, X.; Oler, E.; Li, K.; Paccoud, M.; Hong, C.; Guo, A. C.; Chan, C.; Wei, W.; Ramirez-Gaona, M. PathBank: A Comprehensive Pathway Database for Model Organisms. *Nucleic Acids Res.* **2020**, *48* (D1), D470−D478.
(93) Hafner, J.; MohammadiPeyhani, H.; Sveshnikova, A.; Scheidegger, A.; Hatzimanikatis, V. Updated ATLAS of Biochemistry with New Metabolites and Improved Enzyme Prediction Power. *ACS Synth. Biol.* **2020**, *9* (6), 1479−1482.
(94) Ganter, M.; Bernard, T.; Moretti, S.; Stelling, J.; Pagni, M. Metanetx.org: A Website and Repository for Accessing, Analysing and Manipulating Metabolic Networks. *Bioinformatics* **2013**, *29* (6), 815−816.
(95) Bairoch, A. The ENZYME Database in 2000. *Nucleic Acids Res.* **2000**, *28* (1), 304−305.
(96) McDonald, A. G.; Tipton, K. F. Enzyme Nomenclature and Classification: The State of the Art. *FEBS J.* **2023**, *290* (9), 2214−2231.
(97) Probst, D.; Manica, M.; Nana Teukam, Y. G.; Castrogiovanni, A.; Paratore, F.; Laino, T. Biocatalysed Synthesis Planning Using Data-Driven Learning. *Nat. Commun.* **2022**, *13*, 964.
(98) Heid, E.; Probst, D.; Green, W. H.; Madsen, G. K. H. EnzymeMap: Curation, Validation and Data-Driven Prediction of Enzymatic Reactions. *ChemRxiv* **2023**, DOI: 10.26434/chemrxiv-2023-jzw9w.
(99) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583−589.
(100) Bileschi, M. L.; Belanger, D.; Bryant, D. H.; Sanderson, T.; Carter, B.; Sculley, D.; Bateman, A.; DePristo, M. A.; Colwell, L. J. Using Deep Learning to Annotate the Protein Universe. *Nat. Biotechnol.* **2022**, *40* (6), 932−937.
(101) Nallapareddy, V.; Bordin, N.; Sillitoe, I.; Heinzinger, M.; Littmann, M.; Waman, V. P.; Sen, N.; Rost, B.; Orengo, C. CATHe: Detection of Remote Homologues for CATH Superfamilies Using Embeddings from Protein Language Models. *Bioinformatics* **2023**, *39*, No. btad029.
(102) Jiang, S.-Y.; Jin, J.; Sarojam, R.; Ramachandran, S. A Comprehensive Survey on the Terpene Synthase Gene Family Provides New Insight into Its Evolutionary Patterns. *Genome Biol. Evol.* **2019**, *11* (8), 2078−2098.
(103) Claudel-Renard, C.; Chevalet, C.; Faraut, T.; Kahn, D. Enzyme-Specific Profiles for Genome Annotation: PRIAM. *Nucleic Acids Res.* **2003**, *31* (22), 6633−6639.
(104) Shen, H.-B.; Chou, K.-C. EzyPred: A Top−down Approach for Predicting Enzyme Functional Classes and Subclasses. *Biochem. Biophys. Res. Commun.* **2007**, *364* (1), 53−59.
(105) Dalkiran, A.; Rifaioglu, A. S.; Martin, M. J.; Cetin-Atalay, R.; Atalay, V.; Doğan, T. ECPred: A Tool for the Prediction of the Enzymatic Functions of Protein Sequences Based on the EC Nomenclature. *BMC Bioinformatics* **2018**, *19*, 334.
(106) Huang, W.-L.; Chen, H.-M.; Hwang, S.-F.; Ho, S.-Y. Accurate Prediction of Enzyme Subfamily Class Using an Adaptive Fuzzy K-Nearest Neighbor Method. *Biosystems.* **2007**, *90* (2), 405−413.
(107) Nasibov, E.; Kandemir-Cavas, C. Efficiency Analysis of KNN and Minimum Distance-Based Classifiers in Enzyme Family Prediction. *Comput. Biol. Chem.* **2009**, *33* (6), 461−464.
(108) De Ferrari, L.; Aitken, S.; van Hemert, J.; Goryanin, I. EnzML: Multi-Label Prediction of Enzyme Classes Using InterPro Signatures. *BMC Bioinformatics* **2012**, *13*, 61.

(109) Dobson, P. D.; Doig, A. J. Predicting Enzyme Class from Protein Structure without Alignments. *J. Mol. Biol.* **2005**, *345* (1), 187−199.
(110) Kumar, N.; Skolnick, J. EFICAz2.5: Application of a High-Precision Enzyme Function Predictor to 396 Proteomes. *Bioinformatics* **2012**, *28* (20), 2687−2688.
(111) Matsuta, Y.; Ito, M.; Tohsato, Y. ECOH: An Enzyme Commission Number Predictor Using Mutual Information and a Support Vector Machine. *Bioinformatics* **2013**, *29* (3), 365−372.
(112) Li, Y. H.; Xu, J. Y.; Tao, L.; Li, X. F.; Li, S.; Zeng, X.; Chen, S. Y.; Zhang, P.; Qin, C.; Zhang, C.; Chen, Z.; Zhu, F.; Chen, Y. Z. SVM-Prot 2016: A Web-Server for Machine Learning Prediction of Protein Functional Families from Sequence Irrespective of Similarity. *PLoS One* **2016**, *11* (8), No. e0155290.
(113) Nagao, C.; Nagano, N.; Mizuguchi, K. Prediction of Detailed Enzyme Functions and Identification of Specificity Determining Residues by Random Forests. *PLoS One* **2014**, *9* (1), No. e84623.
(114) Kumar, C.; Choudhary, A. A Top-down Approach to Classify Enzyme Functional Classes and Sub-Classes Using Random Forest. *EURASIP J. Bioinform. Syst. Biol.* **2012**, *2012*, 1.
(115) Volpato, V.; Adelfio, A.; Pollastri, G. Accurate Prediction of Protein Enzymatic Class by N-to-1 Neural Networks. *BMC Bioinformatics* **2013**, *14*, S11.
(116) Amidi, A.; Amidi, S.; Vlachakis, D.; Megalooikonomou, V.; Paragios, N.; Zacharaki, E. I. EnzyNet: Enzyme Classification Using 3D Convolutional Neural Networks on Spatial Representation. *PeerJ* **2018**, *6*, No. e4750.
(117) Ryu, J. Y.; Kim, H. U.; Lee, S. Y. Deep Learning Enables High-Quality and High-Throughput Prediction of Enzyme Commission Numbers. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116* (28), 13996−14001.
(118) Sanderson, T.; Bileschi, M. L.; Belanger, D.; Colwell, L. J. ProteInfer, Deep Neural Networks for Protein Functional Inference. *Elife* **2023**, *12*, No. e80942, DOI: 10.7554/eLife.80942.
(119) Yu, T.; Cui, H.; Li, J. C.; Luo, Y.; Jiang, G.; Zhao, H. Enzyme Function Prediction Using Contrastive Learning. *Science* **2023**, *379* (6639), 1358−1363.
(120) Levin, I.; Liu, M.; Voigt, C. A.; Coley, C. W. Merging Enzymatic and Synthetic Chemistry with Computational Synthesis Planning. *Nat. Commun.* **2022**, *13*, 7747.
(121) Zheng, S.; Zeng, T.; Li, C.; Chen, B.; Coley, C. W.; Yang, Y.; Wu, R. Deep Learning Driven Biosynthetic Pathways Navigation for Natural Products with BioNavi-NP. *Nat. Commun.* **2022**, *13*, 3342.
(122) Watanabe, N.; Yamamoto, M.; Murata, M.; Vavricka, C. J.; Ogino, C.; Kondo, A.; Araki, M. Comprehensive Machine Learning Prediction of Extensive Enzymatic Reactions. *J. Phys. Chem. B* **2022**, *126* (36), 6762−6770.
(123) Kroll, A.; Ranjan, S.; Engqvist, M. K. M.; Lercher, M. J. A General Model to Predict Small Molecule Substrates of Enzymes Based on Machine and Deep Learning. *Nat. Commun.* **2023**, *14*, 2787.
(124) Goldman, S.; Das, R.; Yang, K. K.; Coley, C. W. Machine Learning Modeling of Family Wide Enzyme-Substrate Specificity Screens. *PLoS Comput. Biol.* **2022**, *18* (2), No. e1009853.
(125) Berman, H. M.; Gabanyi, M. J.; Kouranov, A.; Micallef, D. I.; Westbrook, J.; Protein Structure Initiative network of investigators Protein Structure Initiative - TargetTrack 2000-2017 - all data files. Zenodo, 2017. DOI: 10.5281/zenodo.821654
(126) Jarzab, A.; Kurzawa, N.; Hopf, T.; Moerch, M.; Zecha, J.; Leijten, N.; Bian, Y.; Musiol, E.; Maschberger, M.; Stoehr, G.; Becher, I.; Daly, C.; Samaras, P.; Mergner, J.; Spanier, B.; Angelov, A.; Werner, T.; Bantscheff, M.; Wilhelm, M.; Klingenspor, M.; Lemeer, S.; Liebl, W.; Hahne, H.; Savitski, M. M.; Kuster, B. Meltome Atlas-Thermal Proteome Stability across the Tree of Life. *Nat. Methods* **2020**, *17* (5), 495−503.
(127) Yang, Y.; Zhao, J.; Zeng, L.; Vihinen, M. ProTstab2 for Prediction of Protein Thermal Stabilities. *Int. J. Mol. Sci.* **2022**, *23* (18), 10798.
(128) Sapoval, N.; Aghazadeh, A.; Nute, M. G.; Antunes, D. A.; Balaji, A.; Baraniuk, R.; Barberan, C. J.; Dannenfelser, R.; Dun, C.; Edrisi, M.; Elworth, R. A. L.; Kille, B.; Kyrillidis, A.; Nakhleh, L.; Wolfe, C. R.; Yan, Z.; Yao, V.; Treangen, T. J. Current Progress and Open Challenges for Applying Deep Learning across the Biosciences. *Nat. Commun.* **2022**, *13*, 1728.
(129) Diaz, D. J.; Kulikova, A. V.; Ellington, A. D.; Wilke, C. O. Using Machine Learning to Predict the Effects and Consequences of Mutations in Proteins. *Curr. Opin. Struct. Biol.* **2023**, *78*, No. 102518.
(130) Thumuluri, V.; Martiny, H.-M.; Almagro Armenteros, J. J.; Salomon, J.; Nielsen, H.; Johansen, A. R. NetSolP: Predicting Protein Solubility in Escherichia Coli Using Language Models. *Bioinformatics* **2022**, *38* (4), 941−946.
(131) Caldararu, O.; Mehra, R.; Blundell, T. L.; Kepp, K. P. Systematic Investigation of the Data Set Dependency of Protein Stability Predictors. *J. Chem. Inf. Model.* **2020**, *60* (10), 4772−4784.
(132) Mazurenko, S. Predicting Protein Stability and Solubility Changes upon Mutations: Data Perspective. *ChemCatChem* **2020**, *12* (22), 5590−5598.
(133) Velecký, J.; Hamsikova, M.; Stourac, J.; Musil, M.; Damborsky, J.; Bednar, D.; Mazurenko, S. SoluProtMutDB: A Manually Curated Database of Protein Solubility Changes upon Mutations. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 6339−6347.
(134) Wang, S.; Tang, H.; Zhao, Y.; Zuo, L. BayeStab: Predicting Effects of Mutations on Protein Stability with Uncertainty Quantification. *Protein Sci.* **2022**, *31* (11), No. e4467.
(135) Nikam, R.; Kulandaisamy, A.; Harini, K.; Sharma, D.; Gromiha, M. M. ProThermDB: Thermodynamic Database for Proteins and Mutants Revisited after 15 Years. *Nucleic Acids Res.* **2021**, *49* (D1), D420−D424.
(136) Iqbal, S.; Ge, F.; Li, F.; Akutsu, T.; Zheng, Y.; Gasser, R. B.; Yu, D.-J.; Webb, G. I.; Song, J. PROST: AlphaFold2-Aware Sequence-Based Predictor to Estimate Protein Stability Changes upon Missense Mutations. *J. Chem. Inf. Model.* **2022**, *62* (17), 4270−4282.
(137) Hernández, I. M.; Dehouck, Y.; Bastolla, U.; López-Blanco, J. R.; Chacón, P. Predicting Protein Stability Changes upon Mutation Using a Simple Orientational Potential. *Bioinformatics* **2023**, *39* (1), No. btad011, DOI: 10.1093/bioinformatics/btad011.
(138) Xavier, J. S.; Nguyen, T.-B.; Karmarkar, M.; Portelli, S.; Rezende, P. M.; Velloso, J. P. L.; Ascher, D. B.; Pires, D. E. V. ThermoMutDB: A Thermodynamic Database for Missense Mutations. *Nucleic Acids Res.* **2021**, *49* (D1), D475−D479.
(139) Pak, M. A.; Markhieva, K. A.; Novikova, M. S.; Petrov, D. S.; Vorobyev, I. S.; Maksimova, E. S.; Kondrashov, F. A.; Ivankov, D. N. Using AlphaFold to Predict the Impact of Single Mutations on Protein Stability and Function. *PLoS One* **2023**, *18* (3), No. e0282689.
(140) Tsuboyama, K.; Dauparas, J.; Chen, J.; Laine, E.; Mohseni Behbahani, Y.; Weinstein, J. J.; Mangan, N. M.; Ovchinnikov, S.; Rocklin, G. J. Mega-Scale Experimental Analysis of Protein Folding Stability in Biology and Design. *Nature* **2023**, *620* (7973), 434−444.
(141) Yang, Y.; Zeng, L.; Vihinen, M. PON-Sol2: Prediction of Effects of Variants on Protein Solubility. *Int. J. Mol. Sci.* **2021**, *22* (15), 8027.
(142) Li, F.; Yuan, L.; Lu, H.; Li, G.; Chen, Y.; Engqvist, M. K. M.; Kerkhoven, E. J.; Nielsen, J. Deep Learning-Based Kcat Prediction Enables Improved Enzyme-Constrained Model Reconstruction. *Nature Catalysis* **2022**, *5* (8), 662−672.
(143) Xie, W. J.; Asadi, M.; Warshel, A. Enhancing Computational Enzyme Design by a Maximum Entropy Strategy. *Proc. Natl. Acad. Sci. U. S. A.* **2022**, *119* (7), No. e2122355119, DOI: 10.1073/pnas.2122355119.
(144) Ostafe, R.; Fontaine, N.; Frank, D.; Ng Fuk Chong, M.; Prodanovic, R.; Pandjaitan, R.; Offmann, B.; Cadet, F.; Fischer, R. One-Shot Optimization of Multiple Enzyme Parameters: Tailoring Glucose Oxidase for PH and Electron Mediators. *Biotechnol. Bioeng.* **2020**, *117* (1), 17−29.
(145) Høie, M. H.; Cagiada, M.; Beck Frederiksen, A. H.; Stein, A.; Lindorff-Larsen, K. Predicting and Interpreting Large-Scale Mutagenesis Data Using Analyses of Protein Stability and Conservation. *Cell Rep.* **2022**, *38* (2), No. 110207.

(146) Cendrowska, J. PRISM: An Algorithm for Inducing Modular Rules. *Int. J. Man. Mach. Stud.* **1987**, *27* (4), 349−370.
(147) Gupta, A.; Agrawal, S. Machine Learning-Based Enzyme Engineering of PETase for Improved Efficiency in Degrading Non-Biodegradable Plastic. *bioRxiv* **2022**, DOI: 10.1101/2022.01.11.475766.
(148) Gado, J. E.; Beckham, G. T.; Payne, C. M. Improving Enzyme Optimum Temperature Prediction with Resampling Strategies and Ensemble Learning. *J. Chem. Inf. Model.* **2020**, *60* (8), 4098−4107.
(149) Voutilainen, S.; Heinonen, M.; Andberg, M.; Jokinen, E.; Maaheimo, H.; Pääkkönen, J.; Hakulinen, N.; Rouvinen, J.; Lähdesmäki, H.; Kaski, S.; Rousu, J.; Penttilä, M.; Koivula, A. Substrate Specificity of 2-Deoxy-D-Ribose 5-Phosphate Aldolase (DERA) Assessed by Different Protein Engineering and Machine Learning Methods. *Appl. Microbiol. Biotechnol.* **2020**, *104* (24), 10515−10529.
(150) Prabakaran, R.; Rawat, P.; Kumar, S.; Michael Gromiha, M. ANuPP: A Versatile Tool to Predict Aggregation Nucleating Regions in Peptides and Proteins. *J. Mol. Biol.* **2021**, *433* (11), No. 166707.
(151) Thangakani, A. M.; Nagarajan, R.; Kumar, S.; Sakthivel, R.; Velmurugan, D.; Gromiha, M. M. CPAD, Curated Protein Aggregation Database: A Repository of Manually Curated Experimental Data on Protein and Peptide Aggregation. *PLoS One* **2016**, *11* (4), No. e0152949.
(152) Rawat, P.; Prabakaran, R.; Sakthivel, R.; Mary Thangakani, A.; Kumar, S.; Gromiha, M. M. CPAD 2.0: A Repository of Curated Experimental Data on Aggregating Proteins and Peptides. *Amyloid* **2020**, *27* (2), 128−133.
(153) Beerten, J.; Van Durme, J.; Gallardo, R.; Capriotti, E.; Serpell, L.; Rousseau, F.; Schymkowitz, J. WALTZ-DB: A Benchmark Database of Amyloidogenic Hexapeptides. *Bioinformatics* **2015**, *31* (10), 1698−1700.
(154) Louros, N.; Konstantoulea, K.; De Vleeschouwer, M.; Ramakers, M.; Schymkowitz, J.; Rousseau, F. WALTZ-DB 2.0: An Updated Database Containing Structural Information of Experimentally Determined Amyloid-Forming Peptides. *Nucleic Acids Res.* **2020**, *48* (D1), D389−D393.
(155) Wozniak, P. P.; Kotulska, M. AmyLoad: Website Dedicated to Amyloidogenic Protein Fragments. *Bioinformatics* **2015**, *31* (20), 3395−3397.
(156) Liu, X.; Luo, Y.; Li, P.; Song, S.; Peng, J. Deep Geometric Representations for Modeling Effects of Mutations on Protein-Protein Binding Affinity. *PLoS Comput. Biol.* **2021**, *17* (8), No. e1009284.
(157) Jankauskaite, J.; Jiménez-García, B.; Dapkunas, J.; Fernández-Recio, J.; Moal, I. H. SKEMPI 2.0: An Updated Benchmark of Changes in Protein-Protein Binding Energy, Kinetics and Thermodynamics upon Mutation. *Bioinformatics* **2019**, *35* (3), 462−469.
(158) Stourac, J.; Dubrava, J.; Musil, M.; Horackova, J.; Damborsky, J.; Mazurenko, S.; Bednar, D. FireProtDB: Database of Manually Curated Protein Stability Data. *Nucleic Acids Res.* **2021**, *49* (D1), D319−D324.
(159) Pancotti, C.; Benevenuta, S.; Birolo, G.; Alberini, V.; Repetto, V.; Sanavia, T.; Capriotti, E.; Fariselli, P. Predicting Protein Stability Changes upon Single-Point Mutation: A Thorough Comparison of the Available Tools on a New Dataset. *Brief. Bioinform.* **2022**, *23* (2), No. bbab555, DOI: 10.1093/bib/bbab555.
(160) Livesey, B. J.; Marsh, J. A. Updated Benchmarking of Variant Effect Predictors Using Deep Mutational Scanning. *bioRxiv* **2022**, DOI: 10.1101/2022.11.19.517196.
(161) Dunham, A. S.; Beltrao, P. Exploring Amino Acid Functions in a Deep Mutational Landscape. *Mol. Syst. Biol.* **2021**, *17* (7), No. e10305.
(162) Reeb, J.; Wirth, T.; Rost, B. Variant Effect Predictions Capture Some Aspects of Deep Mutational Scanning Experiments. *BMC Bioinformatics* **2020**, *21*, 107.
(163) Gray, V. E.; Hause, R. J.; Luebeck, J.; Shendure, J.; Fowler, D. M. Quantitative Missense Variant Effect Prediction Using Large-Scale Mutagenesis Data. *Cell Syst* **2018**, *6* (1), 116−124.e3.
(164) Notin, P.; Dias, M.; Frazer, J.; Hurtado, J. M.; Gomez, A. N.; Marks, D.; Gal, Y. Tranception: Protein Fitness Prediction with Autoregressive Transformers and Inference-Time Retrieval. In *Proceedings of the 39th International Conference on Machine Learning*; Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S., Eds.; Proceedings of Machine Learning Research, Vol. *162*; PMLR, 2022; pp 16990−17017.
(165) Markin, C. J.; Mokhtari, D. A.; Sunden, F.; Appel, M. J.; Akiva, E.; Longwell, S. A.; Sabatti, C.; Herschlag, D.; Fordyce, P. M. Revealing Enzyme Functional Architecture via High-Throughput Microfluidic Enzyme Kinetics. *Science* **2021**, *373* (6553), eabf8761 DOI: 10.1126/science.abf8761.
(166) Thompson, S.; Zhang, Y.; Ingle, C.; Reynolds, K. A.; Kortemme, T. Altered Expression of a Quality Control Protease in E. Coli Reshapes the in Vivo Mutational Landscape of a Model Enzyme. *Elife* **2020**, *9*, No. e53476, DOI: 10.7554/eLife.53476.
(167) Nikoomanzar, A.; Vallejo, D.; Chaput, J. C. Elucidating the Determinants of Polymerase Specificity by Microfluidic-Based Deep Mutational Scanning. *ACS Synth. Biol.* **2019**, *8* (6), 1421−1429.
(168) Mighell, T. L.; Thacker, S.; Fombonne, E.; Eng, C.; O'Roak, B. J. An Integrated Deep-Mutational-Scanning Approach Provides Clinical Insights on PTEN Genotype-Phenotype Relationships. *Am. J. Hum. Genet.* **2020**, *106* (6), 818−829.
(169) Wang, X.; Zhang, X.; Peng, C.; Shi, Y.; Li, H.; Xu, Z.; Zhu, W. D3DistalMutation: A Database to Explore the Effect of Distal Mutations on Enzyme Activity. *J. Chem. Inf. Model.* **2021**, *61* (5), 2499−2508.
(170) Ma, E. J.; Siirola, E.; Moore, C.; Kummer, A.; Stoeckli, M.; Faller, M.; Bouquet, C.; Eggimann, F.; Ligibel, M.; Huynh, D.; Cutler, G.; Siegrist, L.; Lewis, R. A.; Acker, A.-C.; Freund, E.; Koch, E.; Vogel, M.; Schlingensiepen, H.; Oakeley, E. J.; Snajdrova, R. Machine-Directed Evolution of an Imine Reductase for Activity and Stereoselectivity. *ACS Catal.* **2021**, *11* (20), 12433−12445.
(171) Wu, Z.; Kan, S. B. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine Learning-Assisted Directed Protein Evolution with Combinatorial Libraries. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116* (18), 8852−8858.
(172) Li, G.; Qin, Y.; Fontaine, N. T.; Ng Fuk Chong, M.; Maria-Solano, M. A.; Feixas, F.; Cadet, X. F.; Pandjaitan, R.; Garcia-Borràs, M.; Cadet, F.; Reetz, M. T. Machine Learning Enables Selection of Epistatic Enzyme Mutants for Stability Against Unfolding and Detrimental Aggregation. *Chembiochem* **2021**, *22* (5), 904−914.
(173) Sarkar, A.; Yang, Y.; Vihinen, M. Variation Benchmark Datasets: Update, Criteria, Quality and Applications. *Database* **2020**, *2020*, No. baz117.
(174) Miton, C. M.; Tokuriki, N. How Mutational Epistasis Impairs Predictability in Protein Evolution and Design. *Protein Sci.* **2016**, *25* (7), 1260−1272.
(175) Wittmund, M.; Cadet, F.; Davari, M. D. Learning Epistasis and Residue Coevolution Patterns: Current Trends and Future Perspectives for Advancing Enzyme Engineering. *ACS Catal.* **2022**, *12* (22), 14243−14263.
(176) Yu, H.; Ma, S.; Li, Y.; Dalby, P. A. Hot Spots-Making Directed Evolution Easier. *Biotechnol. Adv.* **2022**, *56*, No. 107926.
(177) Sumbalova, L.; Stourac, J.; Martinek, T.; Bednar, D.; Damborsky, J. HotSpot Wizard 3.0: Web Server for Automated Design of Mutations and Smart Libraries Based on Sequence Input Information. *Nucleic Acids Res.* **2018**, *46* (W1), W356−W362.
(178) Khersonsky, O.; Lipsh, R.; Avizemer, Z.; Ashani, Y.; Goldsmith, M.; Leader, H.; Dym, O.; Rogotner, S.; Trudeau, D. L.; Prilusky, J.; Amengual-Rigo, P.; Guallar, V.; Tawfik, D. S.; Fleishman, S. J. Automated Design of Efficient and Functionally Diverse Enzyme Repertoires. *Mol. Cell* **2018**, *72* (1), 178−186.e5.
(179) Clifton, B. E.; Kozome, D.; Laurino, P. Efficient Exploration of Sequence Space by Sequence-Guided Protein Engineering and Design. *Biochemistry* **2023**, *62* (2), 210−220.
(180) Hie, B. L.; Shanker, V. R.; Xu, D.; Bruun, T. U. J.; Weidenbacher, P. A.; Tang, S.; Wu, W.; Pak, J. E.; Kim, P. S. Efficient

Evolution of Human Antibodies from General Protein Language Models. *Nat. Biotechnol.* **2023**, DOI: 10.1038/s41587-023-01763-2.
(181) Goudy, O. J.; Nallathambi, A.; Kinjo, T.; Randolph, N.; Kuhlman, B. In Silico Evolution of Protein Binders with Deep Learning Models for Structure Prediction and Sequence Design. *bioRxiv* **2023**, DOI: 10.1101/2023.05.03.539278.
(182) Linder, J.; Bogard, N.; Rosenberg, A. B.; Seelig, G. A Generative Neural Network for Maximizing Fitness and Diversity of Synthetic DNA and Protein Sequences. *Cell Syst* **2020**, *11* (1), 49−62.e16.
(183) Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. *arXiv [cs.CV]* **2013**, DOI: 10.48550/arXiv.1312.6199.
(184) Yu, T.; Boob, A. G.; Singh, N.; Su, Y.; Zhao, H. In Vitro Continuous Protein Evolution Empowered by Machine Learning and Automation. *Cell Syst* **2023**, *14*, 633.
(185) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-Learning-Guided Directed Evolution for Protein Engineering. *Nat. Methods* **2019**, *16* (8), 687−694.
(186) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Informed Training Set Design Enables Efficient Machine Learning-Assisted Directed Protein Evolution. *Cell Syst* **2021**, *12* (11), 1026−1045.e7.
(187) Hie, B.; Bryson, B. D.; Berger, B. Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design. *Cell Syst* **2020**, *11* (5), 461−477.e9.
(188) Jain, M.; Deleu, T.; Hartford, J.; Liu, C.-H.; Hernandez-Garcia, A.; Bengio, Y. GFlowNets for AI-Driven Scientific Discovery. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2302.00615.
(189) Bengio, E.; Jain, M.; Korablyov, M.; Precup, D.; Bengio, Y. Flow Network Based Generative Models for Non-Iterative Diverse Candidate Generation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 27381−27394.
(190) Qiu, Y.; Wei, G.-W. CLADE 2.0: Evolution-Driven Cluster Learning-Assisted Directed Evolution. *J. Chem. Inf. Model.* **2022**, *62* (19), 4629−4641.
(191) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified Rational Protein Engineering with Sequence-Based Deep Representation Learning. *Nat. Methods* **2019**, *16* (12), 1315−1322.
(192) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N Protein Engineering with Data-Efficient Deep Learning. *Nat. Methods* **2021**, *18* (4), 389−396.
(193) Hsu, C.; Nisonoff, H.; Fannjiang, C.; Listgarten, J. Learning Protein Fitness Models from Evolutionary and Assay-Labeled Data. *Nat. Biotechnol.* **2022**, *40* (7), 1114−1122.
(194) Zheng, Z.; Deng, Y.; Xue, D.; Zhou, Y.; Fei, Y. E.; Gu, Q. Structure-Informed Language Models Are Protein Designers. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2302.01649.
(195) Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *Life-extension*, 2020. https://life-extension.github.io/2020/05/27/GPT%E6%8A%80%E6%9C%AF%E5%88%9D%E6%8E%A2/language-models.pdf (accessed 2023-06-08).
(196) Harris, Z. S. Distributional Structure. *Word World* **1954**, *10* (2−3), 146−162.
(197) Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; Bhowmik, D.; Rost, B. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44* (10), 7112−7127.
(198) Clifford, J. N.; Høie, M. H.; Deleuran, S.; Peters, B.; Nielsen, M.; Marcatili, P. BepiPred-3.0: Improved B-Cell Epitope Prediction Using Protein Language Models. *Protein Sci.* **2022**, *31* (12), No. e4497.
(199) Elnaggar, A.; Essam, H.; Salah-Eldin, W.; Moustafa, W.; Elkerdawy, M.; Rochereau, C.; Rost, B. Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling. *bioRxiv* **2023**, DOI: 10.1101/2023.01.16.524265.
(200) Pokharel, S.; Pratyush, P.; Heinzinger, M.; Newman, R. H.; Kc, D. B. Improving Protein Succinylation Sites Prediction Using Embeddings from Protein Language Model. *Sci. Rep.* **2022**, *12*, 16933.
(201) Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research, Vol. *97*; PMLR, 2019; pp 2790−2799.
(202) Yang, W.; Liu, C.; Li, Z. Lightweight Fine-Tuning a Pretrained Protein Language Model for Protein Secondary Structure Prediction. *bioRxiv (Bioengineering)*, March 23, 2023, 2023.03.22.530066, ver. 1. DOI: 10.1101/2023.03.22.530066.
(203) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H. UniProt Consortium. UniRef Clusters: A Comprehensive and Scalable Alternative for Improving Sequence Similarity Searches. *Bioinformatics* **2015**, *31* (6), 926−932.
(204) Nijkamp, E.; Ruffolo, J.; Weinstein, E. N.; Naik, N.; Madani, A. ProGen2: Exploring the Boundaries of Protein Language Models. *arXiv [cs.LG]* **2022**, DOI: 10.48550/arXiv.2206.13517.
(205) Finn, R. D.; Bateman, A.; Clements, J.; Coggill, P.; Eberhardt, R. Y.; Eddy, S. R.; Heger, A.; Hetherington, K.; Holm, L.; Mistry, J.; Sonnhammer, E. L. L.; Tate, J.; Punta, M. Pfam: The Protein Families Database. *Nucleic Acids Res.* **2014**, *42*, D222−D230.
(206) Joosten, R. P.; Salzemann, J.; Bloch, V.; Stockinger, H.; Berglund, A.-C.; Blanchet, C.; Bongcam-Rudloff, E.; Combet, C.; Da Costa, A. L.; Deleage, G.; Diarena, M.; Fabbretti, R.; Fettahi, G.; Flegel, V.; Gisel, A.; Kasam, V.; Kervinen, T.; Korpelainen, E.; Mattila, K.; Pagni, M.; Reichstadt, M.; Breton, V.; Tickle, I. J.; Vriend, G. PDB_REDO: Automated Re-Refinement of X-Ray Structure Models in the PDB. *J. Appl. Crystallogr.* **2009**, *42*, 376−384.
(207) Dauparas, J.; Anishchenko, I.; Bennett, N.; Bai, H.; Ragotte, R. J.; Milles, L. F.; Wicky, B. I. M.; Courbet, A.; de Haas, R. J.; Bethel, N.; Leung, P. J. Y.; Huddy, T. F.; Pellock, S.; Tischer, D.; Chan, F.; Koepnick, B.; Nguyen, H.; Kang, A.; Sankaran, B.; Bera, A. K.; King, N. P.; Baker, D. Robust Deep Learning−Based Protein Sequence Design Using ProteinMPNN. *Science* **2022**, *378* (6615), 49−56.
(208) Sillitoe, I.; Lewis, T. E.; Cuff, A.; Das, S.; Ashford, P.; Dawson, N. L.; Furnham, N.; Laskowski, R. A.; Lee, D.; Lees, J. G.; Lehtinen, S.; Studer, R. A.; Thornton, J.; Orengo, C. A. CATH: Comprehensive Structural and Functional Annotations for Genome Sequences. *Nucleic Acids Res.* **2015**, *43*, D376−D381.
(209) Varadi, M.; Anyango, S.; Deshpande, M.; Nair, S.; Natassia, C.; Yordanova, G.; Yuan, D.; Stroe, O.; Wood, G.; Laydon, A.; Žídek, A.; Green, T.; Tunyasuvunakool, K.; Petersen, S.; Jumper, J.; Clancy, E.; Green, R.; Vora, A.; Lutfi, M.; Figurnov, M.; Cowie, A.; Hobbs, N.; Kohli, P.; Kleywegt, G.; Birney, E.; Hassabis, D.; Velankar, S. AlphaFold Protein Structure Database: Massively Expanding the Structural Coverage of Protein-Sequence Space with High-Accuracy Models. *Nucleic Acids Res.* **2022**, *50* (D1), D439−D444.
(210) Rao, R. M.; Liu, J.; Verkuil, R.; Meier, J.; Canny, J.; Abbeel, P.; Sercu, T.; Rives, A. MSA Transformer. In *Proceedings of the 38th International Conference on Machine Learning*; Meila, M., Zhang, T., Eds.; Proceedings of Machine Learning Research, Vol. *139*; PMLR, 2021; pp 8844−8856.
(211) Ho, J.; Kalchbrenner, N.; Weissenborn, D.; Salimans, T. Axial Attention in Multidimensional Transformers. *arXiv [cs.CV]* **2019**, DOI: 10.48550/arXiv.1912.12180.
(212) Repecka, D.; Jauniskis, V.; Karpus, L.; Rembeza, E.; Rokaitis, I.; Zrimec, J.; Poviloniene, S.; Laurynenas, A.; Viknander, S.; Abuajwa, W.; Savolainen, O.; Meskys, R.; Engqvist, M. K. M.; Zelezniak, A. Expanding Functional Protein Sequence Spaces Using Generative Adversarial Networks. *Nature Machine Intelligence* **2021**, *3* (4), 324−333.
(213) Sevgen, E.; Moller, J.; Lange, A.; Parker, J.; Quigley, S.; Mayer, J.; Srivastava, P.; Gayatri, S.; Hosfield, D.; Korshunova, M.; Livne, M.; Gill, M.; Ranganathan, R.; Costa, A. B.; Ferguson, A. L. ProT-VAE: Protein Transformer Variational AutoEncoder for Functional Protein

Design. *bioRxiv (Synthetic Biology)*, January 24, 2023, 2023.01.23.525232, ver. 1. DOI: 10.1101/2023.01.23.525232.
(214) Luo, Y.; Jiang, G.; Yu, T.; Liu, Y.; Vo, L.; Ding, H.; Su, Y.; Qian, W. W.; Zhao, H.; Peng, J. ECNet Is an Evolutionary Context-Integrated Deep Learning Framework for Protein Engineering. *Nat. Commun.* **2021**, *12*, 5743.
(215) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9* (8), 1735−1780.
(216) Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; Millán, C.; Park, H.; Adams, C.; Glassman, C. R.; DeGiovanni, A.; Pereira, J. H.; Rodrigues, A. V.; van Dijk, A. A.; Ebrecht, A. C.; Opperman, D. J.; Sagmeister, T.; Buhlheller, C.; Pavkov-Keller, T.; Rathinaswamy, M. K.; Dalwadi, U.; Yip, C. K.; Burke, J. E.; Garcia, K. C.; Grishin, N. V.; Adams, P. D.; Read, R. J.; Baker, D. Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network. *Science* **2021**, *373* (6557), 871−876.
(217) Illig, A.-M.; Siedhoff, N. E.; Schwaneberg, U.; Davari, M. D. A Hybrid Model Combining Evolutionary Probability and Machine Learning Leverages Data-Driven Protein Engineering. *bioRxiv* **2022**, DOI: 10.1101/2022.06.07.495081.
(218) Ding, X.; Zou, Z.; Brooks, C. L., Iii Deciphering Protein Evolution and Fitness Landscapes with Latent Space Models. *Nat. Commun.* **2019**, *10*, 5644.
(219) Kohout, P.; Vasina, M.; Majerova, M.; Novakova, V.; Damborsky, J.; Bednar, D.; et al. Design of Enzymes for Biocatalysis, Bioremediation and Biosensing Using Variational Autoencoder-Generated Latent Space. ChemRxiv. Cambridge: Cambridge Open Engage, 2023. DOI: 10.26434/chemrxiv-2023-jcds7.
(220) Ziegler, C.; Martin, J.; Sinner, C.; Morcos, F. Latent Generative Landscapes as Maps of Functional Diversity in Protein Sequence Space. *Nat. Commun.* **2023**, *14*, 2222.
(221) Moffat, L.; Jones, D. T. Increasing the Accuracy of Single Sequence Prediction Methods Using a Deep Semi-Supervised Learning Framework. *Bioinformatics* **2021**, *37* (21), 3744−3751.
(222) Bepler, T.; Berger, B. Learning Protein Sequence Embeddings Using Information from Structure. *International Conference on Learning Representations*, New Orleans, LA, May 6−9, 2019; OpenReview, 2019. https://openreview.net/forum?id=SygLehCqtm
(223) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating Protein Transfer Learning with TAPE. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9689−9701.
(224) Crean, R. M.; Gardner, J. M.; Kamerlin, S. C. L. Harnessing Conformational Plasticity to Generate Designer Enzymes. *J. Am. Chem. Soc.* **2020**, *142* (26), 11324−11342.
(225) Guo, H.-B.; Perminov, A.; Bekele, S.; Kedziora, G.; Farajollahi, S.; Varaljay, V.; Hinkle, K.; Molinero, V.; Meister, K.; Hung, C.; Dennis, P.; Kelley-Loughnane, N.; Berry, R. AlphaFold2 Models Indicate That Protein Sequence Determines Both Structure and Dynamics. *Sci. Rep.* **2022**, *12*, 10696.
(226) Faidon Brotzakis, Z.; Zhang, S.; Vendruscolo, M. AlphaFold Prediction of Structural Ensembles of Disordered Proteins. *bioRxiv (Biophysics)*, January 19, 2023, 2023.01.19.524720, ver. 1. DOI: 10.1101/2023.01.19.524720.
(227) Piana, S.; Laio, A. Advillin Folding Takes Place on a Hypersurface of Small Dimensionality. *Phys. Rev. Lett.* **2008**, *101* (20), No. 208101.
(228) Glielmo, A.; Husic, B. E.; Rodriguez, A.; Clementi, C.; Noé, F.; Laio, A. Unsupervised Learning Methods for Molecular Simulation Data. *Chem. Rev.* **2021**, *121* (16), 9722−9758.
(229) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. VAMPnets for Deep Learning of Molecular Kinetics. *Nat. Commun.* **2018**, *9*, 5.
(230) Marques, S. M.; Kouba, P.; Legrand, A.; Sedlar, J.; Disson, L.; Planas-Iglesias, J.; Sanusi, Z.; Kunka, A.; Damborsky, J.; Pajdla, T.; Prokop, Z.; Mazurenko, S.; Sivic, J.; Bednar, D. Effects of Alzheimer's Disease Drug Candidates on Disordered Aβ42 Dissected by Comparative Markov State Analysis (CoVAMPnet). *bioRxiv (Biophysics)*, January 6, **2023**, 2023.01.06.523007, ver. 1. DOI: 10.1101/2023.01.06.523007.
(231) Ward, M. D.; Zimmerman, M. I.; Meller, A.; Chung, M.; Swamidass, S. J.; Bowman, G. R. Deep Learning the Structural Determinants of Protein Biochemical Properties by Comparing Structural Ensembles with DiffNets. *Nat. Commun.* **2021**, *12*, 3023.
(232) Akere, A.; Chen, S. H.; Liu, X.; Chen, Y.; Dantu, S. C.; Pandini, A.; Bhowmik, D.; Haider, S. Structure-Based Enzyme Engineering Improves Donor-Substrate Recognition of Arabidopsis Thaliana Glycosyltransferases. *Biochem. J.* **2020**, *477* (15), 2791−2805.
(233) Russ, W. P.; Figliuzzi, M.; Stocker, C.; Barrat-Charlaix, P.; Socolich, M.; Kast, P.; Hilvert, D.; Monasson, R.; Cocco, S.; Weigt, M.; Ranganathan, R. An Evolution-Based Model for Designing Chorismate Mutase Enzymes. *Science* **2020**, *369* (6502), 440−445.
(234) Lu, H.; Diaz, D. J.; Czarnecki, N. J.; Zhu, C.; Kim, W.; Shroff, R.; Acosta, D. J.; Alexander, B. R.; Cole, H. O.; Zhang, Y.; Lynd, N. A.; Ellington, A. D.; Alper, H. S. Machine Learning-Aided Engineering of Hydrolases for PET Depolymerization. *Nature* **2022**, *604* (7907), 662−667.
(235) Paik, I.; Ngo, P. H. T.; Shroff, R.; Diaz, D. J.; Maranhao, A. C.; Walker, D. J. F.; Bhadra, S.; Ellington, A. D. Improved Bst DNA Polymerase Variants Derived via a Machine Learning Approach. *Biochemistry* **2023**, *62* (2), 410−418.
(236) Weinstein, J. J.; Goldenzweig, A.; Hoch, S.; Fleishman, S. J. PROSS 2: A New Server for the Design of Stable and Highly Expressed Protein Variants. *Bioinformatics* **2021**, *37* (1), 123−125.
(237) Musil, M.; Stourac, J.; Bendl, J.; Brezovsky, J.; Prokop, Z.; Zendulka, J.; Martinek, T.; Bednar, D.; Damborsky, J. FireProt: Web Server for Automated Design of Thermostable Proteins. *Nucleic Acids Res.* **2017**, *45* (W1), W393−W399.
(238) Kunka, A.; Marques, S.; Havlasek, M.; Vasina, M.; Velatova, N.; Cengelova, L.; Kovar, D.; Damborsky, J.; Marek, M.; Bednar, D.; Prokop, Z. Advancing Enzyme′s Stability and Catalytic Efficiency through Synergy of Force-Field Calculations, Evolutionary Analysis and Machine Learning. *ACS Catal.* **2023**, *13*, 12506−12518.
(239) Wicky, B. I. M.; Milles, L. F.; Courbet, A.; Ragotte, R. J.; Dauparas, J.; Kinfu, E.; Tipps, S.; Kibler, R. D.; Baek, M.; DiMaio, F.; Li, X.; Carter, L.; Kang, A.; Nguyen, H.; Bera, A. K.; Baker, D. Hallucinating Symmetric Protein Assemblies. *Science* **2022**, *378* (6615), 56−61.
(240) Hawkins-Hooker, A.; Depardieu, F.; Baur, S.; Couairon, G.; Chen, A.; Bikard, D. Generating Functional Protein Variants with Variational Autoencoders. *PLoS Comput. Biol.* **2021**, *17* (2), No. e1008736.
(241) Vasina, M.; Vanacek, P.; Hon, J.; Kovar, D.; Faldynova, H.; Kunka, A.; Buryska, T.; Badenhorst, C. P. S.; Mazurenko, S.; Bednar, D.; Stavrakis, S.; Bornscheuer, U. T.; deMello, A.; Damborsky, J.; Prokop, Z. Advanced Database Mining of Efficient Haloalkane Dehalogenases by Sequence and Structure Bioinformatics and Microfluidics. *Chem. Catalysis* **2022**, *2* (10), 2704−2725.
(242) Pardo, I.; Bednar, D.; Calero, P.; Volke, D. C.; Damborský, J.; Nikel, P. I. A Nonconventional Archaeal Fluorinase Identified by In Silico Mining for Enhanced Fluorine Biocatalysis. *ACS Catal.* **2022**, *12* (11), 6570−6577.
(243) Yeh, A. H.-W.; Norn, C.; Kipnis, Y.; Tischer, D.; Pellock, S. J.; Evans, D.; Ma, P.; Lee, G. R.; Zhang, J. Z.; Anishchenko, I.; Coventry, B.; Cao, L.; Dauparas, J.; Halabiya, S.; DeWitt, M.; Carter, L.; Houk, K. N.; Baker, D. De Novo Design of Luciferases Using Deep Learning. *Nature* **2023**, *614* (7949), 774−780.
(244) Büchler, J.; Malca, S. H.; Patsch, D.; Voss, M.; Turner, N. J.; Bornscheuer, U. T.; Allemann, O.; Le Chapelain, C.; Lumbroso, A.; Loiseleur, O.; Buller, R. Algorithm-Aided Engineering of Aliphatic Halogenase WelO5* for the Asymmetric Late-Stage Functionalization of Soraphens. *Nat. Commun.* **2022**, *13*, 371.
(245) Saito, Y.; Oikawa, M.; Sato, T.; Nakazawa, H.; Ito, T.; Kameda, T.; Tsuda, K.; Umetsu, M. Machine-Learning-Guided Library Design Cycle for Directed Evolution of Enzymes: The Effects

of Training Data Composition on Sequence Space Exploration. *ACS Catal.* **2021**, *11* (23), 14615−14624.
(246) Greenhalgh, J. C.; Fahlberg, S. A.; Pfleger, B. F.; Romero, P. A. Machine Learning-Guided Acyl-ACP Reductase Engineering for Improved in Vivo Fatty Alcohol Production. *Nat. Commun.* **2021**, *12*, 5825.
(247) Schenkmayerova, A.; Pinto, G. P.; Toul, M.; Marek, M.; Hernychova, L.; Planas-Iglesias, J.; Daniel Liskova, V.; Pluskal, D.; Vasina, M.; Emond, S.; Dörr, M.; Chaloupkova, R.; Bednar, D.; Prokop, Z.; Hollfelder, F.; Bornscheuer, U. T.; Damborsky, J. Engineering the Protein Dynamics of an Ancestral Luciferase. *Nat. Commun.* **2021**, *12*, 3616.
(248) Chaloupkova, R.; Liskova, V.; Toul, M.; Markova, K.; Sebestova, E.; Hernychova, L.; Marek, M.; Pinto, G. P.; Pluskal, D.; Waterman, J.; Prokop, Z.; Damborsky, J. Light-Emitting Dehalogenases: Reconstruction of Multifunctional Biocatalysts. *ACS Catal.* **2019**, *9* (6), 4810−4823.
(249) Klesmith, J. R.; Bacik, J.-P.; Wrenbeck, E. E.; Michalczyk, R.; Whitehead, T. A. Trade-Offs between Enzyme Fitness and Solubility Illuminated by Deep Mutational Scanning. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114* (9), 2265−2270.
(250) MacLeod, B. P.; Parlane, F. G. L.; Rupnow, C. C.; Dettelbach, K. E.; Elliott, M. S.; Morrissey, T. D.; Haley, T. H.; Proskurin, O.; Rooney, M. B.; Taherimakhsousi, N.; Dvorak, D. J.; Chiu, H. N.; Waizenegger, C. E. B.; Ocean, K.; Mokhtari, M.; Berlinguette, C. P. A Self-Driving Laboratory Advances the Pareto Front for Material Properties. *Nat. Commun.* **2022**, *13*, 995.
(251) Li, W.; Yao, X.; Zhang, T.; Wang, R.; Wang, L. Hierarchy Ranking Method for Multimodal Multi-Objective Optimization with Local Pareto Fronts. *IEEE Trans. Evol. Computat.* **2023**, *27*, 98.
(252) Miton, C. M.; Tokuriki, N. Insertions and Deletions (Indels): A Missing Piece of the Protein Engineering Jigsaw. *Biochemistry* **2023**, *62* (2), 148−157.
(253) Gonzalez, C. E.; Roberts, P.; Ostermeier, M. Fitness Effects of Single Amino Acid Insertions and Deletions in TEM-1 β-Lactamase. *J. Mol. Biol.* **2019**, *431* (12), 2320−2330.
(254) Fan, X.; Pan, H.; Tian, A.; Chung, W. K.; Shen, Y. SHINE: Protein Language Model-Based Pathogenicity Prediction for Short Inframe Insertion and Deletion Variants. *Brief. Bioinform.* **2023**, *24* (1), No. bbac584, DOI: 10.1093/bib/bbac584.
(255) Ross, C. M.; Foley, G.; Boden, M.; Gillam, E. M. J. Using the Evolutionary History of Proteins to Engineer Insertion-Deletion Mutants from Robust, Ancestral Templates Using Graphical Representation of Ancestral Sequence Predictions (GRASP). *Methods Mol. Biol.* **2022**, *2397*, 85−110.
(256) Park, H.-S.; Nam, S.-H.; Lee, J. K.; Yoon, C. N.; Mannervik, B.; Benkovic, S. J.; Kim, H.-S. Design and Evolution of New Catalytic Activity with an Existing Protein Scaffold. *Science* **2006**, *311* (5760), 535−538.
(257) Babkova, P.; Sebestova, E.; Brezovsky, J.; Chaloupkova, R.; Damborsky, J. Ancestral Haloalkane Dehalogenases Show Robustness and Unique Substrate Specificity. *Chembiochem* **2017**, *18* (14), 1448−1456.
(258) Arpino, J. A. J.; Rizkallah, P. J.; Jones, D. D. Structural and Dynamic Changes Associated with Beneficial Engineered Single-Amino-Acid Deletion Mutations in Enhanced Green Fluorescent Protein. *Acta Crystallogr. D Biol. Crystallogr.* **2014**, *70* (8), 2152−2162.
(259) Dumas, A.; Lercher, L.; Spicer, C. D.; Davis, B. G. Designing Logical Codon Reassignment - Expanding the Chemistry in Biology. *Chem. Sci.* **2015**, *6* (1), 50−69.
(260) Hankore, E. D.; Zhang, L.; Chen, Y.; Liu, K.; Niu, W.; Guo, J. Genetic Incorporation of Noncanonical Amino Acids Using Two Mutually Orthogonal Quadruplet Codons. *ACS Synth. Biol.* **2019**, *8* (5), 1168−1174.
(261) An, X.; Chen, C.; Wang, T.; Huang, A.; Zhang, D.; Han, M.-J.; Wang, J. Genetic Incorporation of Selenotyrosine Significantly Improves Enzymatic Activity of Agrobacterium Radiobacter Phosphotriesterase. *Chembiochem* **2021**, *22* (15), 2535−2539.
(262) Zhang, H.; Zheng, Z.; Dong, L.; Shi, N.; Yang, Y.; Chen, H.; Shen, Y.; Xia, Q. Rational Incorporation of Any Unnatural Amino Acid into Proteins by Machine Learning on Existing Experimental Proofs. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 4930−4941.
(263) Gainza, P.; Sverrisson, F.; Monti, F.; Rodolà, E.; Boscaini, D.; Bronstein, M. M.; Correia, B. E. Deciphering Interaction Fingerprints from Protein Molecular Surfaces Using Geometric Deep Learning. *Nat. Methods* **2020**, *17* (2), 184−192.
(264) Ketata, M. A.; Laue, C.; Mammadov, R.; Stark, H.; Wu, M.; Corso, G.; Marquet, C.; Barzilay, R.; Jaakkola, T. S. DiffDock-PP: Rigid Protein-Protein Docking with Diffusion Models. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, May 1−5, 2023; OpenReview, **2023**. https://openreview.net/pdf?id=AM7WbQxuRS
(265) Geng, C.; Xue, L. C.; Roel-Touris, J.; Bonvin, A. M. J. J. Finding the ΔΔG Spot: Are Predictors of Binding Affinity Changes upon Mutations in Protein−Protein Interactions Ready for It? *WIREs Comput. Mol. Sci.* **2019**, *9* (5), No. e1410.
(266) Jiang, Y.; Quan, L.; Li, K.; Li, Y.; Zhou, Y.; Wu, T.; Lyu, Q. DGCddG: Deep Graph Convolution for Predicting Protein-Protein Binding Affinity Changes Upon Mutations. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2023**, *20* (3), 2089−2100.
(267) Shan, S.; Luo, S.; Yang, Z.; Hong, J.; Su, Y.; Ding, F.; Fu, L.; Li, C.; Chen, P.; Ma, J.; Shi, X.; Zhang, Q.; Berger, B.; Zhang, L.; Peng, J. Deep Learning Guided Optimization of Human Antibody against SARS-CoV-2 Variants with Broad Neutralization. *Proc. Natl. Acad. Sci. U. S. A.* **2022**, *119* (11), No. e2122954119.
(268) Jin, W.; Sarkizova, S.; Chen, X.; Hacohen, N.; Uhler, C. Unsupervised Protein-Ligand Binding Energy Prediction via Neural Euler's Rotation Equation. *arXiv [q-bio.BM]* **2023**, DOI: 10.48550/arXiv.2301.10814.
(269) Jiang, Y.; Neti, S. S.; Sitarik, I.; Pradhan, P.; To, P.; Xia, Y.; Fried, S. D.; Booker, S. J.; O'Brien, E. P. How Synonymous Mutations Alter Enzyme Structure and Function over Long Timescales. *Nat. Chem.* **2023**, *15* (3), 308−318.
(270) Nikolados, E.-M.; Oyarzún, D. A. Deep Learning for Optimization of Protein Expression. *Curr. Opin. Biotechnol.* **2023**, *81*, No. 102941.
(271) Rosenberg, A. A.; Marx, A.; Bronstein, A. M. Codon-Specific Ramachandran Plots Show Amino Acid Backbone Conformation Depends on Identity of the Translated Codon. *Nat. Commun.* **2022**, *13*, 2815.
(272) Saunders, R.; Deane, C. M. Synonymous Codon Usage Influences the Local Protein Structure Observed. *Nucleic Acids Res.* **2010**, *38* (19), 6719−6728.
(273) Outeiral, C.; Deane, C. M. Codon Language Embeddings Provide Strong Signals for Protein Engineering. *bioRxiv* **2022**, DOI: 10.1101/2022.12.15.519894.
(274) Constant, D. A.; Gutierrez, J. M.; Sastry, A. V.; Viazzo, R.; Smith, N. R.; Hossain, J.; Spencer, D. A.; Carter, H.; Ventura, A. B.; Louie, M. T. M.; Kohnert, C.; Consbruck, R.; Bennett, J.; Crawford, K. A.; Sutton, J. M.; Morrison, A.; Steiger, A. K.; Jackson, K. A.; Stanton, J. T.; Abdulhaqq, S.; Hannum, G.; Meier, J.; Weinstock, M.; Gander, M. Deep Learning-Based Codon Optimization with Large-Scale Synonymous Variant Datasets Enables Generalized Tunable Protein Expression. *bioRxiv (Synthetic Biology)*, February 12, 2023, 2023.02.11.528149, ver. 1. DOI: 10.1101/2023.02.11.528149.
(275) Ruscio, J. Z.; Kohn, J. E.; Ball, K. A.; Head-Gordon, T. The Influence of Protein Dynamics on the Success of Computational Enzyme Design. *J. Am. Chem. Soc.* **2009**, *131* (39), 14111−14115.
(276) Peccati, F.; Alunno-Rufini, S.; Jiménez-Osés, G. Accurate Prediction of Enzyme Thermostabilization with Rosetta Using AlphaFold Ensembles. *J. Chem. Inf. Model.* **2023**, *63* (3), 898−909.
(277) Acevedo-Rocha, C. G.; Li, A.; D'Amore, L.; Hoebenreich, S.; Sanchis, J.; Lubrano, P.; Ferla, M. P.; Garcia-Borràs, M.; Osuna, S.; Reetz, M. T. Pervasive Cooperative Mutational Effects on Multiple Catalytic Enzyme Traits Emerge via Long-Range Conformational Dynamics. *Nat. Commun.* **2021**, *12*, 1621.

(278) Bonk, B. M.; Weis, J. W.; Tidor, B. Machine Learning Identifies Chemical Characteristics That Promote Enzyme Catalysis. *J. Am. Chem. Soc.* **2019**, *141* (9), 4108−4118.
(279) Zhong, E. D.; Bepler, T.; Berger, B.; Davis, J. H. CryoDRGN: Reconstruction of Heterogeneous Cryo-EM Structures Using Neural Networks. *Nat. Methods* **2021**, *18* (2), 176−185.
(280) Jia, K.; Kilinc, M.; Jernigan, R. L. Functional Protein Dynamics Directly from Sequences. *J. Phys. Chem. B* **2023**, *127* (9), 1914−1921.
(281) Wang, T.; Zhu, J.-Y.; Torralba, A.; Efros, A. A. Dataset Distillation. *arXiv [cs.LG]* **2018**, DOI: 10.48550/arXiv.1811.10959.
(282) Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv [stat.ML]* **2015**, DOI: 10.48550/arXiv.1503.02531.
(283) Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* **2012**, *29* (6), 141−142.
(284) Le, T.-T.-H.; Larasati, H. T.; Prihatno, A. T.; Kim, H. A Review of Dataset Distillation for Deep Learning. In *Proceedings of the 2022 International Conference on Platform Technology and Service (PlatCon)*, Jeju, South Korea, August 22−24, 2022; IEEE, 2022; pp 34−37. DOI: 10.1109/PlatCon55845.2022.9932086
(285) Yu, R.; Liu, S.; Wang, X. Dataset Distillation: A Comprehensive Review. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2301.07014.
(286) Lei, S.; Tao, D. A Comprehensive Survey of Dataset Distillation. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2301.05603.
(287) Abraham, M.; Apostolov, R.; Barnoud, J.; Bauer, P.; Blau, C.; Bonvin, A. M. J. J.; Chavent, M.; Chodera, J.; Čondić-Jurkić, K.; Delemotte, L.; Grubmüller, H.; Howard, R. J.; Jordan, E. J.; Lindahl, E.; Ollila, O. H. S.; Selent, J.; Smith, D. G. A.; Stansfeld, P. J.; Tiemann, J. K. S.; Trellet, M.; Woods, C.; Zhmurov, A. Sharing Data from Molecular Simulations. *J. Chem. Inf. Model.* **2019**, *59* (10), 4093−4099.
(288) Serafeim, A.-P.; Salamanos, G.; Patapati, K. K.; Glykos, N. M. Sensitivity of Folding Molecular Dynamics Simulations to Even Minor Force Field Changes. *J. Chem. Inf. Model.* **2016**, *56* (10), 2035−2041.
(289) Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E.; Bouwman, J.; Brookes, A. J.; Clark, T.; Crosas, M.; Dillo, I.; Dumon, O.; Edmunds, S.; Evelo, C. T.; Finkers, R.; Gonzalez-Beltran, A.; Gray, A. J. G.; Groth, P.; Goble, C.; Grethe, J. S.; Heringa, J.; 't Hoen, P. A. C.; Hooft, R.; Kuhn, T.; Kok, R.; Kok, J.; Lusher, S. J.; Martone, M. E.; Mons, A.; Packer, A. L.; Persson, B.; Rocca-Serra, P.; Roos, M.; van Schaik, R.; Sansone, S.-A.; Schultes, E.; Sengstag, T.; Slater, T.; Strawn, G.; Swertz, M. A.; Thompson, M.; van der Lei, J.; van Mulligen, E.; Velterop, J.; Waagmeester, A.; Wittenburg, P.; Wolstencroft, K.; Zhao, J.; Mons, B. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Sci. Data* **2016**, *3*, No. 160018.
(290) Tiemann, J. K. S.; Szczuka, M.; Bouarroudj, L.; Oussaren, M.; Garcia, S.; Howard, R. J.; Delemotte, L.; Lindahl, E.; Baaden, M.; Lindorff-Larsen, K.; Chavent, M.; Poulain, P. MDverse: Shedding Light on the Dark Matter of Molecular Dynamics Simulations. *bioRxiv* **2023**, DOI: 10.1101/2023.05.02.538537.
(291) Durumeric, A. E. P.; Charron, N. E.; Templeton, C.; Musil, F.; Bonneau, K.; Pasos-Trejo, A. S.; Chen, Y.; Kelkar, A.; Noé, F.; Clementi, C. Machine Learned Coarse-Grained Protein Force-Fields: Are We There Yet? *Curr. Opin. Struct. Biol.* **2023**, *79*, No. 102533.
(292) Beyer, L.; Hénaff, O. J.; Kolesnikov, A.; Zhai, X.; van den Oord, A. Are We Done with ImageNet? *arXiv [cs.CV]* **2020**, DOI: 10.48550/arXiv.2006.07159.
(293) Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303.
(294) Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, June 20−25, 2009; IEEE, 2009; pp 248−255. DOI: 10.1109/CVPR.2009.5206848
(295) LeCun, Y.; Haffner, P.; Bottou, L.; Bengio, Y. Object Recognition with Gradient-Based Learning. In *Shape, Contour and Grouping in Computer Vision*; Springer, 1999; pp 319−345.
(296) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, June 27−30, 2016; IEEE, 2016; pp 770−778. DOI: 10.1109/CVPR.2016.90
(297) Thiyagalingam, J.; Shankar, M.; Fox, G.; Hey, T. Scientific Machine Learning Benchmarks. *Nature Reviews Physics* **2022**, *4* (6), 413−420.
(298) Steinegger, M.; Söding, J. Clustering Huge Protein Sequence Sets in Linear Time. *Nat. Commun.* **2018**, *9*, 2542.
(299) Gao, M.; Skolnick, J. Structural Space of Protein-Protein Interfaces Is Degenerate, Close to Complete, and Highly Connected. *Proc. Natl. Acad. Sci. U. S. A.* **2010**, *107* (52), 22517−22522.
(300) Burra, P. V.; Zhang, Y.; Godzik, A.; Stec, B. Global Distribution of Conformational States Derived from Redundant Models in the PDB Points to Non-Uniqueness of the Protein Structure. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (26), 10505−10510.
(301) Robin, X.; Leemann, M.; Sagasta, A.; Eberhardt, J.; Schwede, T.; Durairaj, J. Automated Benchmarking of Combined Protein Structure and Ligand Conformation Prediction. *Authorea Preprints* **2023**, DOI: 10.22541/au.168382988.85108031/v1.
(302) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48* (12), 4111−4119.
(303) Dallago, C.; Mou, J.; Johnston, K. E.; Wittmann, B.; Bhattacharya, N.; Goldman, S.; Madani, A.; Yang, K. K. FLIP: Benchmark Tasks in Fitness Landscape Inference for Proteins. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, Vol. *1*; Vanschoren, J., Yeung, S., Eds.; Curran Associates, Inc.: Red Hook, NY, 2021.
(304) Morehead, A.; Chen, C.; Sedova, A.; Cheng, J. DIPS-Plus: The Enhanced Database of Interacting Protein Structures for Interface Prediction. *arXiv [q-bio.QM]* **2021**, DOI: 10.48550/arXiv.2106.04362.
(305) Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; Moult, J. Critical Assessment of Methods of Protein Structure Prediction (CASP)-Round XIV. *Proteins* **2021**, *89* (12), 1607−1617.
(306) Janin, J.; Henrick, K.; Moult, J.; Eyck, L. T.; Sternberg, M. J. E.; Vajda, S.; Vakser, I.; Wodak, S. J. Critical Assessment of PRedicted Interactions. CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins* **2003**, *52* (1), 2−9.
(307) Andreoletti, G.; Hoskins, R. A.; Repo, S.; Barsky, D.; Brenner, S. E.; Mult, J.; Participants, C. Abstract 3295: CAGI: The Critical Assessment of Genome Interpretation, a Community Experiment to Evaluate Phenotype Prediction: Implications for Predicting Impact of Variants in Cancer. *Cancer Res.* **2018**, *78*, 3295−3295.
(308) Grešová, K.; Martinek, V.; Čechák, D.; Šimeček, P.; Alexiou, P. Genomic Benchmarks: A Collection of Datasets for Genomic Sequence Classification. *BMC Genomic Data* **2023**, *24*, 25.
(309) Buterez, D.; Janet, J. P.; Kiddle, S. J.; Liò, P. MF-PCBA: Multifidelity High-Throughput Screening Benchmarks for Drug Discovery and Machine Learning. *J. Chem. Inf. Model.* **2023**, *63* (9), 2667−2678.
(310) Walsh, I.; Fishman, D.; Garcia-Gasulla, D.; Titma, T.; Pollastri, G.; Capriotti, E.; Casadio, R.; Capella-Gutierrez, S.; Cirillo, D.; Del Conte, A.; Dimopoulos, A. C.; Del Angel, V. D.; Dopazo, J.; Fariselli, P.; Fernandez, J. M.; Huber, F.; Kreshuk, A.; Lenaerts, T.; Martelli, P. L.; Navarro, A.; Broin, P. O; Pinero, J.; Piovesan, D.; Reczko, M.; Ronzano, F.; Satagopam, V.; Savojardo, C.; Spiwok, V.; Tangaro, M. A.; Tartari, G.; Salgado, D.; Valencia, A.; Zambelli, F.; Harrow, J.; Psomopoulos, F. E.; Tosatto, S. C. E. DOME: Recommendations for Supervised Machine Learning Validation in Biology. *Nat. Methods* **2021**, *18* (10), 1122−1127.

(311) Mirdita, M.; Schütze, K.; Moriwaki, Y.; Heo, L.; Ovchinnikov, S.; Steinegger, M. ColabFold: Making Protein Folding Accessible to All. *Nat. Methods* **2022**, *19* (6), 679−682.
(312) Lee, B. D.; Gitter, A.; Greene, C. S.; Raschka, S.; Maguire, F.; Titus, A. J.; Kessler, M. D.; Lee, A. J.; Chevrette, M. G.; Stewart, P. A.; Britto-Borges, T.; Cofer, E. M.; Yu, K.-H.; Carmona, J. J.; Fertig, E. J.; Kalinin, A. A.; Signal, B.; Lengerich, B. J.; Triche, T. J., Jr; Boca, S. M. Ten Quick Tips for Deep Learning in Biology. *PLoS Comput. Biol.* **2022**, *18* (3), e1009803. **03**
(313) Samek, W.; Müller, K.-R. Towards Explainable Artificial Intelligence. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., Müller, K.-R., Eds.; Springer, 2019; pp 5−22.
(314) Wellawatte, G. P.; Gandhi, H. A.; Seshadri, A.; White, A. D. A Perspective on Explanations of Molecular Prediction Models. *J. Chem. Theory Comput.* **2023**, *19* (8), 2149−2160.
(315) Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; Samek, W. Explainable AI Methods - A Brief Overview. In *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*; Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.-R., Samek, W., Eds.; Springer, 2022; pp 13−38.
(316) Montavon, G.; Binder, A.; Lapuschkin, S.; Samek, W.; Müller, K.-R. Layer-Wise Relevance Propagation: An Overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., Müller, K.-R., Eds.; Springer, 2019; pp 193−209.
(317) van der Zanden, T. C.; Bodlaender, H. L.; Hamers, H. J. M. Efficiently Computing the Shapley Value of Connectivity Games in Low-Treewidth Graphs. *Oper. Res. Int. J.* **2023**, *23*, 6.
(318) Ribeiro, M. T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '16* San Francisco, CA, August 13−17, 2016; Association for Computing Machinery: New York, NY, 2016; pp 1135−1144.
(319) Ivanovs, M.; Kadikis, R.; Ozols, K. Perturbation-Based Methods for Explaining Deep Neural Networks: A Survey. *Pattern Recognit. Lett.* **2021**, *150*, 228−234.
(320) Ma, J.; Yu, M. K.; Fong, S.; Ono, K.; Sage, E.; Demchak, B.; Sharan, R.; Ideker, T. Using Deep Learning to Model the Hierarchical Structure and Function of a Cell. *Nat. Methods* **2018**, *15* (4), 290−298.
(321) Novakovsky, G.; Dexter, N.; Libbrecht, M. W.; Wasserman, W. W.; Mostafavi, S. Obtaining Genetics Insights from Deep Learning via Explainable Artificial Intelligence. *Nat. Rev. Genet.* **2023**, *24* (2), 125−137.
(322) Fortelny, N.; Bock, C. Knowledge-Primed Neural Networks Enable Biologically Interpretable Deep Learning on Single-Cell Sequencing Data. *Genome Biol.* **2020**, *21*, 190.
(323) Nikolados, E.-M.; Wongprommoon, A.; Aodha, O. M.; Cambray, G.; Oyarzún, D. A. Accuracy and Data Efficiency in Deep Learning Models of Protein Expression. *Nat. Commun.* **2022**, *13*, 7755.
(324) Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; Zhu, J. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In *Natural Language Processing and Chinese Computing*; Springer, 2019; pp 563−574.
(325) Shimazaki, T.; Tachikawa, M. Collaborative Approach between Explainable Artificial Intelligence and Simplified Chemical Interactions to Explore Active Ligands for Cyclin-Dependent Kinase 2. *ACS Omega* **2022**, *7* (12), 10372−10381.
(326) Probst, D. Explainable Prediction of Catalysing Enzymes from Reactions Using Multilayer Perceptrons. *bioRxiv (Bioinformatics)*, January 30, **2023**, 2023.01.28.526009, ver. 1. DOI: 10.1101/2023.01.28.526009.
(327) Li, C.; Liu, J.; Chen, J.; Yuan, Y.; Yu, J.; Gou, Q.; Guo, Y.; Pu, X. An Interpretable Convolutional Neural Network Framework for Analyzing Molecular Dynamics Trajectories: A Case Study on Functional States for G-Protein-Coupled Receptors. *J. Chem. Inf. Model.* **2022**, *62* (6), 1399−1410.
(328) Tan, J.; Zhang, Y. ExplainableFold: Understanding AlphaFold Prediction with Explainable AI. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining KDD '23*; Association for Computing Machinery: New York, NY, 2023; pp 2166−2176.
(329) Hoover, B.; Strobelt, H.; Gehrmann, S. ExBERT: A VIsual ANalysis TOol to EXplore LEarned REpresentations in TRansformer MOdels. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*; Association for Computational Linguistics, 2020; pp 187−196.
(330) Ferruz, N.; Höcker, B. Controllable Protein Design with Language Models. *Nature Machine Intelligence* **2022**, *4* (6), 521−532.
(331) Abd Elrahman, S. M.; Abraham, A. A review of class imbalance problem. *J. Netw. Innov. Comput.* **2013**, *1*, 332−340.
(332) Haixiang, G.; Yijing, L.; Shang, J.; Mingyun, G.; Yuanyue, H.; Bing, G. Learning from Class-Imbalanced Data: Review of Methods and Applications. *Expert Syst. Appl.* **2017**, *73*, 220−239.
(333) Kaur, H.; Pannu, H. S.; Malhi, A. K. A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Comput. Surv.* **2020**, *52* (4), 1−36.
(334) Esposito, C.; Landrum, G. A.; Schneider, N.; Stiefl, N.; Riniker, S. GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning. *J. Chem. Inf. Model.* **2021**, *61* (6), 2623−2640.
(335) Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; Kenton, Z.; Brown, S.; Hawkins, W.; Stepleton, T.; Biles, C.; Birhane, A.; Haas, J.; Rimell, L.; Hendricks, L. A.; Isaac, W.; Legassick, S.; Irving, G.; Gabriel, I. Ethical and Social Risks of Harm from Language Models. *arXiv (Computer Science.Computation and Language)*, December 8, **2021**, 2112.04359, ver. 1. DOI: 10.48550/arXiv.2112.04359
(336) Kessler, M. D.; Yerges-Armstrong, L.; Taub, M. A.; Shetty, A. C.; Maloney, K.; Jeng, L. J. B.; Ruczinski, I.; Levin, A. M.; Williams, L. K.; Beaty, T. H.; Mathias, R. A.; Barnes, K. C.; et al. Challenges and Disparities in the Application of Personalized Genomic Medicine to Populations with African Ancestry. *Nat. Commun.* **2016**, *7*, 12521.
(337) Sullivan, B. J.; Nguyen, T.; Durani, V.; Mathur, D.; Rojas, S.; Thomas, M.; Syu, T.; Magliery, T. J. Stabilizing Proteins from Sequence Statistics: The Interplay of Conservation and Correlation in Triosephosphate Isomerase Stability. *J. Mol. Biol.* **2012**, *420* (4−5), 384−399.
(338) Fang, J. A Critical Review of Five Machine Learning-Based Algorithms for Predicting Protein Stability Changes upon Mutation. *Brief. Bioinform.* **2020**, *21* (4), 1285−1292.
(339) Pucci, F.; Bernaerts, K. V.; Kwasigroch, J. M.; Rooman, M. Quantification of Biases in Predictions of Protein Stability Changes upon Mutations. *Bioinformatics* **2018**, *34* (21), 3659−3665.
(340) Caldararu, O.; Blundell, T. L.; Kepp, K. P. A Base Measure of Precision for Protein Stability Predictors: Structural Sensitivity. *BMC Bioinformatics* **2021**, *22*, 88.
(341) Scantlebury, J.; Vost, L.; Carbery, A.; Hadfield, T. E.; Turnbull, O. M.; Brown, N.; Chenthamarakshan, V.; Das, P.; Grosjean, H.; von Delft, F.; Deane, C. M. A Small Step Toward Generalizability: Training a Machine Learning Scoring Function for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2023**, *63* (10), 2960−2974.
(342) Hebert-Johnson, U.; Kim, M.; Reingold, O.; Rothblum, G. Multicalibration: Calibration for the (COmputationally-Identifiable) Masses. In *Proceedings of the 35th International Conference on Machine Learning*; Dy, J., Krause, A., Eds.; Proceedings of Machine Learning Research; PMLR, 10--15 Jul 2018; Vol. *80*, pp 1939−1948.
(343) Gopalan, P.; Kim, M. P.; Singhal, M. A.; Zhao, S. Low-Degree Multicalibration. In *Proceedings of Thirty Fifth Conference on Learning Theory*; Loh, P.-L., Raginsky, M., Eds.; Proceedings of Machine Learning Research, Vol. *178*; PMLR, 2022; pp 3193−3234.

(344) Kim, M. P.; Ghorbani, A.; Zou, J. Multiaccuracy: Black-Box Post-Processing for Fairness in Classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society AIES '19*; Association for Computing Machinery: New York, NY, 2019; pp 247−254.
(345) Pessach, D.; Shmueli, E. Algorithmic Fairness. *arXiv [cs.CY]* **2020**, DOI: 10.48550/arXiv.2001.09784.
(346) Minot, M.; Reddy, S. T. Meta Learning Improves Robustness and Performance in Machine Learning-Guided Protein Engineering. *bioRxiv*, January 30, 2023, 2023.01.30.526201, ver. 1. DOI: 10.1101/2023.01.30.526201.
(347) Musaelian, A.; Johansson, A.; Batzner, S.; Kozinsky, B. Scaling the Leading Accuracy of Deep Equivariant Models to Biomolecular Simulations of Realistic Size. *arXiv [physics.comp-ph]* **2023**, DOI: 10.48550/arXiv.2304.10061.
(348) Shaw, D. E.; Adams, P. J.; Azaria, A.; Bank, J. A.; Batson, B.; Bell, A.; Bergdorf, M.; Bhatt, J.; Butts, J. A.; Correia, T.; Dirks, R. M.; Dror, R. O.; Eastwood, M. P.; Edwards, B.; Even, A.; Feldmann, P.; Fenn, M.; Fenton, C. H.; Forte, A.; Gagliardo, J.; Gill, G.; Gorlatova, M.; Greskamp, B.; Grossman, J. P.; Gullingsrud, J.; Harper, A.; Hasenplaugh, W.; Heily, M.; Heshmat, B. C.; Hunt, J.; Ierardi, D. J.; Iserovich, L.; Jackson, B. L.; Johnson, N. P.; Kirk, M. M.; Klepeis, J. L.; Kuskin, J. S.; Mackenzie, K. M.; Mader, R. J.; McGowen, R.; McLaughlin, A.; Moraes, M. A.; Nasr, M. H.; Nociolo, L. J.; O'Donnell, L.; Parker, A.; Peticolas, J. L.; Pocina, G.; Predescu, C.; Quan, T.; Salmon, J. K.; Schwink, C.; Shim, K. S.; Siddique, N.; Spengler, J.; Szalay, T.; Tabladillo, R.; Tartler, R.; Taube, A. G.; Theobald, M.; Towles, B.; Vick, W.; Wang, S. C.; Wazlowski, M.; Weingarten, M. J.; Williams, J. M.; Yuh, K. A. Anton 3: Twenty Microseconds of Molecular Dynamics Simulation before Lunch. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis SC '21*; Association for Computing Machinery: New York, NY, 2021; pp 1−11.
(349) Perdomo-Ortiz, A.; Benedetti, M.; Realpe-Gómez, J.; Biswas, R. Opportunities and Challenges for Quantum-Assisted Machine Learning in near-Term Quantum Computers. *Quantum Sci. Technol.* **2018**, 3 (3), No. 030502.
(350) Caro, M. C.; Huang, H.-Y.; Cerezo, M.; Sharma, K.; Sornborger, A.; Cincio, L.; Coles, P. J. Generalization in Quantum Machine Learning from Few Training Data. *Nat. Commun.* **2022**, *13*, 4919.
(351) Daley, A. J.; Bloch, I.; Kokail, C.; Flannigan, S.; Pearson, N.; Troyer, M.; Zoller, P. Practical Quantum Advantage in Quantum Simulation. *Nature* **2022**, *607* (7920), 667−676.
(352) Ollitrault, P. J.; Miessen, A.; Tavernelli, I. Molecular Quantum Dynamics: A Quantum Computing Perspective. *Acc. Chem. Res.* **2021**, *54* (23), 4229−4238.
(353) Bender, E. M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency FAccT '21*; Association for Computing Machinery: New York, NY, 2021; pp 610−623.
(354) Patterson, D.; Gonzalez, J.; Holzle, U.; Le, Q.; Liang, C.; Munguia, L.-M.; Rothchild, D.; So, D. R.; Texier, M.; Dean, J. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer* **2022**, *55*, 18.
(355) Vinod, R.; Chen, P.-Y.; Das, P. Reprogramming Pretrained Language Models for Protein Sequence Representation Learning. *arXiv [cs.LG]* **2023**, DOI: 10.48550/arXiv.2301.02120.
(356) Caldararu, O.; Blundell, T. L.; Kepp, K. P. Three Simple Properties Explain Protein Stability Change upon Mutation. *J. Chem. Inf. Model.* **2021**, *61* (4), 1981−1988.
(357) Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations*, April 25−29, 2022; OpenReview, 2022. https://openreview.net/forum?id=nZeVKeeFYf9
(358) Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C. Stanford Alpaca: An Instruction-Following Llama Model. 2023.
(359) Yang, A.; Miech, A.; Sivic, J.; Laptev, I.; Schmid, C.; Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; Oh, A. Zero-Shot Video Question Answering via Frozen Bidirectional Language Models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 124−141.
(360) Anstine, D. M.; Isayev, O. Generative Models as an Emerging Paradigm in the Chemical Sciences. *J. Am. Chem. Soc.* **2023**, *145* (16), 8736−8750.
(361) Popova, M.; Isayev, O.; Tropsha, A. Deep Reinforcement Learning for de Novo Drug Design. *Sci Adv* **2018**, *4* (7), No. eaap7885.
(362) Lutz, I. D.; Wang, S.; Norn, C.; Courbet, A.; Borst, A. J.; Zhao, Y. T.; Dosey, A.; Cao, L.; Xu, J.; Leaf, E. M.; Treichel, C.; Litvicov, P.; Li, Z.; Goodson, A. D.; Rivera-Sánchez, P.; Bratovianu, A.-M.; Baek, M.; King, N. P.; Ruohola-Baker, H.; Baker, D. Top-down Design of Protein Architectures with Reinforcement Learning. *Science* **2023**, *380* (6642), 266−273.
(363) Wang, Y.; Tang, H.; Huang, L.; Pan, L.; Yang, L.; Yang, H.; Mu, F.; Yang, M. Self-Play Reinforcement Learning Guides Protein Engineering. *Nature Machine Intelligence* **2023**, *5* (8), 845−860.

https://doi.org/10.1021/acscatal.3c02743
ACS Catal. 2023, 13, 13863−13895