Guiding Generative Models for Protein Design: Prompting, Steering and Aligning

Filippo Stocco[1,2,◇], Michele Garibbo[1,◇] and Noelia Ferruz[1,2,*]

[1]Centre for Genomic Regulation, the Barcelona Institute of Science and Technology,

Dr Aiguader 88, Barcelona 08003, Spain

[2]Universitat Pompeu Fabra (UPF), Barcelona, Spain

[◇]equally contributed

Emails: {filippo.stocco, michele.garibbo, noelia.ferruz} @crg.eu

August 2025

Abstract

Generative artificial intelligence (AI) models learn probability distributions from data and produce novel samples that capture the salient properties of their training sets. Proteins are particularly attractive for such approaches given their abundant data and the versatility of their representations, ranging from sequences to structures and functions. This versatility has motivated the rapid development of generative models for protein design, enabling the generation of functional proteins and enzymes with unprecedented success. However, because these models mirror their training distribution, they tend to sample from its most probable modes, while low-probability regions, often encoding valuable properties, remain underexplored. To address this challenge, recent work has focused on guiding generative models to produce proteins with user-specified properties, even when such properties are rare or absent from the original training distribution. In this review, we survey and categorize recent advances in conditioning generative models for protein design. We distinguish approaches that modify model parameters, such as reinforcement learning or supervised fine-tuning, from those that keep the model fixed, including conditional generation, retrieval-augmented strategies, Bayesian guidance, and tailored sampling methods. Together, these developments are beginning to enable the steering of generative models toward proteins with desired, and often previously inaccessible, properties.

1 Introduction

Learning to generate samples from complex probability distributions lies at the core of modern generative modeling. In the context of proteins, the availability of large datasets has catalyzed the rapid development of powerful generative models for protein design (GMPDs). Among GMPDs, some of the most adopted architectures are diffusion models, which reconstruct atomic coordinates by reversing a noise-adding process (Fig. 1A)[1, 2, 3], and protein language models (pLMs), trained either on masked sequence reconstruction (e.g. ESM[4] or the MSA Transformer[5]) or on autoregressive next-token prediction (e.g. ProtGPT2[6], ProGen2-3[7], and Evo-1/2[8, 9]) (Fig. 1B). In recent years, such models have enabled remarkable achievements, including the design of binders [10], ligand-binding receptors [11], and *de novo* enzymes [12] on unprecedented timescales.

Trained in a large corpus of natural sequences, structures, and functional annotations [e.g., 13, 14, 7, 15], GMPDs can explore vast regions of the protein sequence–structure landscape [e.g., 7, 16, 17, 18]. However, protein engineering often targets exceptional properties that are rare or even disfavored by natural selection (such as extreme thermostability in mesophilic organisms). These desirable functional optima may correspond to isolated high-fitness peaks separated by deep valleys in the evolutionary landscape (Fig. 1C, D), representing trajectories that evolution is unlikely to traverse. Consequently, GMPDs trained solely on natural data tend to assign negligible probability to these regions, making them difficult to sample directly. Further biases arise from uneven phylogenetic representation and residual annotation errors in the underlying datasets [e.g., 19, 20, 21, 22], which may further distort the learned landscape. In this sense, recent work highlight that simply increasing model scale does not guarantee monotonic improvements in fitness prediction [23].

However, when appropriately guided, GMPDs have the potential to access low-probability yet functionally optimal regions of protein space. More formally, the protein design objective can be defined as generating protein sequences x with desired properties y, where y may represent attributes such as thermostability, catalytic efficiency, binding specificity, or other design goals. By contrast, the primary training objective of a GMPD is to model the distribution of natural proteins, $p(x)$. To bridge this gap, recent approaches aim to model the conditional distribution $p(x|y)$, thereby biasing generation toward sequences with the desired attributes (Fig. 1C). In many cases, this objective can also be viewed as an optimization problem, where the goal is to find x that maximizes $p(x|y)$. Several recent techniques can be conceptually unified under this $p(x|y)$ framework, showing strong success in aligning GMPDs with specific design objectives. In this review, we categorize these strategies into two broad classes: (i) train-time methods, which modify the model parameters so that the learned distribution
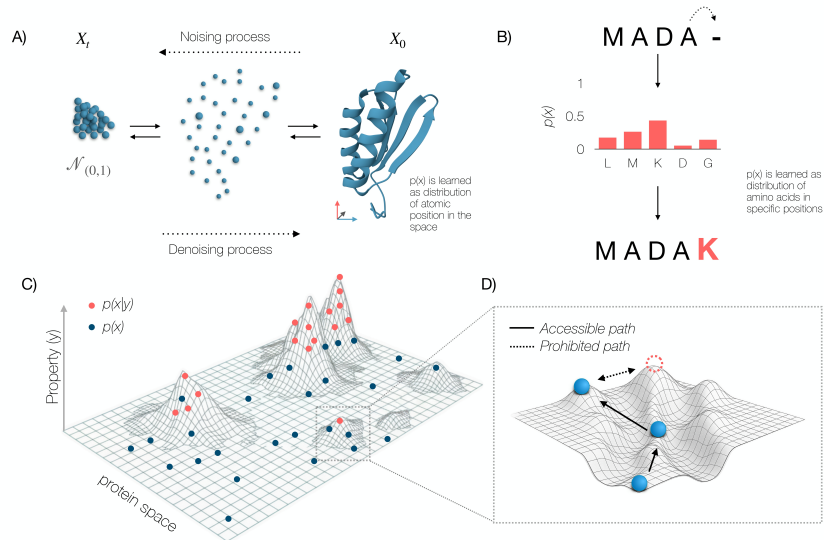
Figure 1: **Generative modeling approaches and schematic illustrations of protein fitness landscapes.** (A) *Diffusion-based generative models.* Protein structures are represented by the distribution $p(x)$ of atomic coordinates in three-dimensional space. During training, true atomic coordinates X_0 are progressively corrupted by Gaussian noise until reaching a normal distribution $\mathcal{N}(0, 1)$. At inference, the model reverses this process, iteratively denoising random noise to reconstruct a protein structure. (B) *Sequence-based generative models.* Here, $p(x)$ denotes the distribution of amino acids across sequence positions. Training can proceed via masked-token prediction, in which the model infers the identity of masked residues, or via autoregressive next-token prediction, where sequences are generated one residue at a time. (C) *Protein fitness landscape.* Schematic illustration of the relationship between the data distribution $p(x)$ (blue points), typically learned in an unsupervised manner, and the conditional distribution $p(y|x)$ (red points), which focuses sampling on regions associated with high fitness. (D) *Evolutionary accessibility.* Evolution explores the fitness landscape through local, incremental mutations that can reach only contiguous high-fitness regions (solid paths), whereas transitions across fitness valleys (dashed paths) are inaccessible. In contrast, generative protein design models (GMPDs) can, in principle, traverse the landscape in a less constrained manner, directly sampling from otherwise evolutionarily inaccessible regions.

$p(x)$ approximates $p(x|y)$, and (ii) inference-time control, which guides generation toward target properties without altering the model's underlying weights.

1.1 Train-time methods for controlled Generation

Train-time methods modify the underlying probability distribution by directly updating GMPDs' parameters. The most straightforward approach in this category is Supervised Fine-Tuning (SFT). In SFT, a pre-trained GMPD is further optimized under the same objective used during pre-training, but on a curated dataset of high-quality examples. For instance, a pre-trained protein language model (pLM) can be fine-tuned on a carefully assembled dataset from a target enzyme family to generate novel and distant members of that family. This process adapts the model's parameters to a specific domain, effectively shifting its generative prior to align with the target data. SFT has achieved notable success, including the generation of enzymes [24, 25], gene editors[26, 27], or bacteriophages [28].

While SFT effectively specializes GMPDs toward generating samples representative of a particular dataset, it does not provide the model with the ability to discriminate by data quality, i.e., to differentiate among varying degrees of a desired property. Similar limitations have been observed in Natural Language Processing (NLP), where SFT alone often leads to suboptimal alignment with user intent and [29], in some cases, catastrophic forgetting [30]. Consequently, SFT is now commonly combined with reinforcement learning (RL) to achieve finer control over model behavior [e.g 31].

In RL, a model learns to make optimal decisions by interacting with an environment through trial and error, receiving feedback in the form of rewards or penalties to maximize its cumulative reward over time. Unlike SFT, in RL the model is not provided with explicit examples of the desired outputs. Instead, the GMPD must infer and explore autonomously, potentially uncovering novel solutions that might not have been anticipated. More technically, a pre-trained model is treated as a policy $\pi_\theta(x)$ and updated to maximize a scalar reward -or equivalently, the probability over preferences- while constraining excessive deviation from its pre-trained distribution. Similarly to SFT, this process enables a transition from the unconditional distribution $p(x)$ toward the desired conditional distribution $p(x|y)$. Today, RL is central to the alignment of large language models (LLMs) and has driven remarkable advances across diverse fields, from autonomous driving to game playing.

Several RL techniques have been implemented over the years. REINVENT (2017) [32] provides an early attempt, in the molecular field, to leverage vanilla RL policy-gradient methods (i.e., REINFORCE) to move from a broad distribution $p(x)$ toward a desired conditional distribution $p(x|y)$. REINVENT re-frames the popular REINFORCE update as an "augmented-likelihood objective". This objective allows to increase a property y's score, while retaining probability mass near the original distribution $p(x)$, allowing the steering sam-

	Method	Description	Examples
Train-time methods	Supervised finetuning (SFT)	Fit the model to well curated data, shifting the learned distribution towards the data.	–
	Reinforcement Learning	Aligns the model on feedback data over model's outputs. Methods include preference- or reward-driven learning.	PPO, GRPO, DPO
Inference-time control	Prompt & context programming	Generation guided by structuring the input prompt with explicit instructions or templates (e.g., for specific positions or motifs).	Masked pLM, EvoDiff, BoltzGen
	Retrieval-Augmented Generation (RAG)	Enhances generation by dynamically incorporating external knowledge retrieved from a large corpus.	RAG pipelines
	Output-dependent guidance	Gradient-based guided generation in sequence space based on the inference output.	PPLM, ColabDesign, BoltzDesign, BindCraft
	Activation steering	Direct manipulation of hidden states (e.g., residual stream) to promote or suppress attributes without parameter updates.	Sparse Autoencoders (SAEs), steering vectors
	Bayesian guidance	Re-weights the probability distribution using Bayes' theorem: $\tilde{p}(y \mid x) \propto p_\theta(y \mid x) \exp(\lambda s(y, x))$.	Bayesian scoring functions
	Sampling controls	Alters sampling strategy (temperature, top-k, top-p, or more advanced searches like beam search and MCTS) to influence randomness and diversity.	MCTS, temperature scaling, top-k, top-p

Table 1: Overview of Train-time methods and Inference-time Control Methods with Examples. pLM: protein language model. MCTS: Monte Carlo Tree Search.

ples into high-scoring regions without sacrificing realism. Shortly after, Proximal Policy Approximation (PPO, 2017) [33] was introduced, providing a key improvement over vanilla policy-gradient methods; its clipped surrogate objective approximates a trust region, guaranteeing more stable updates.
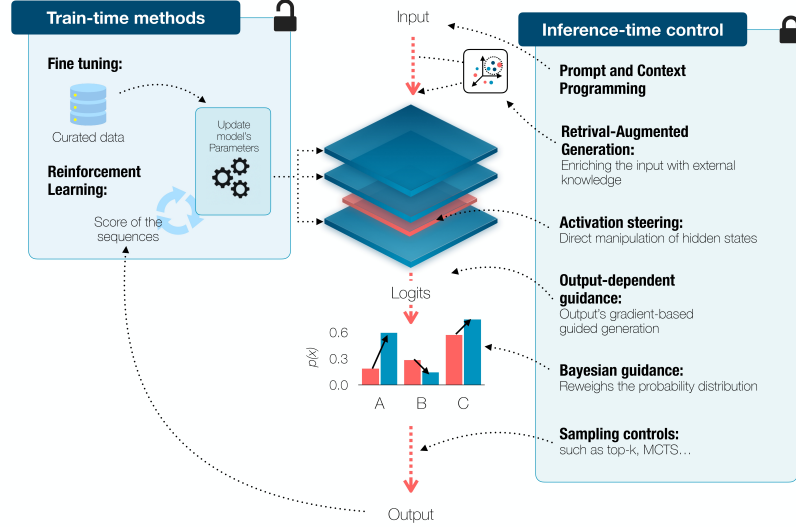
Figure 2: Different stages where intervention can occur.

PPO became the basis for one of the two main families of RL from Human Feedback (RLFH, 2022) approaches, allowing to align LLM to human preferences. In the first approach, a reward model is first trained on (human) ranked responses (i.e., preferences) for desired properties y (e.g., clarity of text, non-offensive text). Subsequently, the reward model is used within the PPO update to align a pre-trained LLM, encoding $p(x)$, towards the properties y. Similarly to REINVENT, RLHF augments the reward with a penalty for drifting too far from the original distribution $p(x)$. The more recent GRPO algorithm [34] (2024) computes the rewards for groups of LLM's responses, augmenting the PPO update with a reward baseline (i.e., computed across the group), without needing to train any additional value model. Direct Preference Optimization [35] (DPO, 2023) introduced the second family of RLHF approaches. DPO exploits the same preference signal as PPO-based approaches, but recasts alignment as a supervised objective on the log-probability differences between ranked (human) preferences. In practice, this dispenses the need to train an explicit reward model, provides a simpler and, potentially, more stable route toward preference alignment.

These techniques have quickly met considerable success in the protein research realm, as summarized in table 2. PPO-style functional alignment has been used to update pLMs with experimental measurements, e.g., RL from eXperimental Feedback (RLXF) aligns an ESM-based generator toward brighter CreiLOV variants while constraining drift from the original distribution[36]. Model-based RL in the AlphaZero style has been applied to backbone construction, where

6

Monte-Carlo Tree Search guided by a policy-value network outperforms plain tree search on top-down design tasks[37]. Preference-based tuning with DPO has been used to bias structure-conditioned models toward stability (preferring stabilizing over destabilizing sequences given a target backbone) and to reduce MHC-I epitope load while maintaining the fold[38, 39]. In parallel, mutation-policy RL frameworks propose on-policy sequence edits under specific oracles[36], and ProtRL extends the broad application of DPO and GRPO to protein engineering with pLMs, a framework that led to the design of low-nanomolar EGFR inhibitors[29].

Method	Core objective	Protein-design uses (2022-2025)
REINFORCE (Vanilla Policy Gradient, 1992 [40])	Learns a policy such that actions with higher returns have higher likelihood of being sampled.	AB-gen[41]
PPO (Proximal Policy Optimization, 2017[42])	Plain policy gradients such as REINFORCE are quite noisy and unstable. PPO clips the model's gradient update to not deviate too much from original model distribution.	RLXF[36][43]
AlphaZero MCTS (Monte Carlo Tree Search with policy–value networks, 2018 [44])	A neural networks with two heads is used to predict the most promising actions as well as the final rewards given a current position. This network is used to guide a MCTS to explore the most promising actions efficiently.	EvoPlay[45], HighPlay[46]
DPO (Direct Preference Optimization, [35])	Learns directly from ranked preference data, without requiring an explicit reward model.	ProteinDPO[38], ProtRL[29] and Park et al.[47]
GRPO (Group Relative Policy Optimization, [48])	For each prompt it samples a group of G candidate outputs, scores them with a reward, and uses relative (within-group) advantages to update the policy with a PPO-style clipped loss, without the need of a value model.	ProtRL[29], ProteinZero [43]

Table 2: **Representative reinforcement learning frameworks applied to protein sequence design.** Summary of selected reinforcement learning (RL)–based approaches illustrating how core objectives have been adapted for protein or molecular design tasks between 2022 and 2025. The examples listed here are not exhaustive and focus on policy based RL, but highlight major methodological directions in the recent literature. For more details see [49], and Supplementary.

1.2 Inference-time control

In contrast to train-time methods, *inference-time control* methods guide GM-PDs without updating their parameters, by directly influencing the models at the time of generation. These approaches assume that a pretrained GMPD already captures a rich representation of protein sequence-structure relationships, and that new behaviors can be elicited by steering the model's inference process rather than retraining it. Such methods offer advantages in efficiency and flexibility, as the same model can be repurposed for diverse design tasks, enabling rapid exploration of new hypotheses while avoiding costly retraining.

Interventions can be applied at distinct stages of the generative process, from input to final sampling. Figure 2 summarizes these methodologies, categorized according to the site of intervention. Prompt and context engineering refers to shaping model behavior by modifying its inputs or conditioning information. pLMs can be trained with various control tags, such as Enzyme Commission numbers (ZymCTRL, [24]), Uniprot functional keywords (Progen, [25]), taxonomy (Evo1/2, [8, 9]), or combinations thereof [50]. Upon prompting these labels, the model specifically generates sequences for those cases. In other cases, specific residues may be masked and the model asked to reconstruct them, thereby exploring local sequence neighborhoods, or the model can be conditioned on predefined motifs and catalytic residues to guide functional design [51, 52, 53, 54, 10, 55]. Related to this idea, retrieval-augmented generation (RAG) has been introduced to dynamically incorporate external knowledge into the GMPD's context[56, 57, 58]. By drawing upon semantically related examples from large databases, RAG enables more informed sampling and can enrich the design process with functional or structural priors not explicitly encoded in the pretrained model. A recent example is Protriever[57], which introduces a retrieval-augmented protein language model that jointly learns to retrieve homologous sequences and model their fitness, integrating evolutionary context at inference time without explicit structural supervision. Interventions can be applied at distinct stages of the generative process, from input to final sampling. Figure 2 summarizes these methodologies, categorized according to the site of intervention. Prompt and context engineering refers to shaping model behavior by modifying its inputs or conditioning information. pLMs can be trained with various control tags, such as Enzyme Commission numbers (ZymCTRL, [24]), Uniprot functional keywords (Progen1, [25]), taxonomy (Evo1/2, [8, 9]), or combinations thereof [50]. Upon prompting these labels, the model specifically generates sequences for those cases. In other cases, specific residues may be masked and the model asked to reconstruct them, thereby exploring local sequence neighborhoods[59], or the model can be conditioned on predefined motifs and catalytic residues to guide functional design [51, 60, 53]. Other approaches act directly on the latent encodings of protein sequences in a feedback-optimization loop, exploring regions of protein-structure space that

satisfy user-defined design objectives such as ColabDesign, BindCraft and following methods [10, 54]. Alternatively, architectures such as BoltzGen[3] implement a conditional generative diffusion model with continuous guidance, in which encoded design conditions—such as binding-site specifications or structural constraints—are propagated throughout the denoising process to steer generation toward conformations consistent with the imposed design criteria [61] .

With the aim to dynamically inject additional knowledge into the GMPD's context, Retrieval-augmented generation (RAG) has been successfully applied in different cases[56, 57, 58]. By drawing upon semantically related examples from large databases, RAG enables more informed sampling and can enrich the design process with functional or structural priors not explicitly encoded in the pretrained model. A recent example is Protriever[57], which introduces a retrieval-augmented protein language model that jointly learns to retrieve homologous sequences and model their fitness, integrating evolutionary context at inference time without explicit structural supervision.

A more surgical form of intervention acts directly within the hidden states of the network. Activation steering manipulates internal representations, often within the residual stream, by injecting vectors that correspond to interpretable latent directions. Sparse autoencoders (SAEs) have been used to identify such interpretable features in protein language models, revealing latent dimensions correlated with properties like enzymatic activity, hydrophobicity, or thermostability [62, 63]. For example, Parsan et al. used SAE-derived features to bias structure predictions in ESMFold toward more hydrophobic conformations through feature steering[64], while Boxò et al. leveraged activity-associated features to steer ZymCTRL toward more active α-amylases[65]. This approach provides a surgical way of shaping the trajectory of the model's activations without altering inputs or outputs explicitly.

Control can also be applied at the level of output probabilities. Bayesian guidance reweighs the probability distribution encoded by GMPDs using Bayesian principles, effectively combining the model's prior with external evidence or predictive scores. Such strategies have been applied in protein design, where sequence likelihoods are updated according to functional predictors or activity [66].

Finally, Sampling Controls manipulate the stochasticity of the GMPD's final output by, for example, manipulating "inference parameters" like temperature, top-k and top-p sampling, balancing diversity and fidelity, which is particularly important when sampling from vast sequence landscapes [67]. More advanced sampling techniques, such as beam search and Monte Carlo Tree Search (MCTS), allow to consider multiple GMPDs inference trajectories, selecting the optimal ones [e.g., 9, 68].

Recent theoretical work underscores the generality of these approaches: flow matching in discrete state spaces has been shown to be equivalent to masked language modeling, autoregressive generation, and diffusion. This unifying perspective positions inference-time control as a suite of architecture-agnostic, plug-and-play techniques that can be ported across model classes with minimal modification [66].

1.3   Conclusion and future prospects

Protein design operates at the intersection of evolutionary complexity and computational abstraction. Natural proteins are shaped by diverse and often competing forces: biochemical constraints, ecological pressures, and evolutionary contingencies; making it difficult to define a single, global "fitness vector". Instead, the true fitness landscape is fluid, heterogeneous, and context-dependent. Combined with well-known dataset biases [e.g., see 19, 20, 22], these properties impose severe limitations on how much we can leverage the natural distribution of proteins to design and engineer proteins *à la carte*, with full control over the design process.

Here, we discussed two broad categories of methods, fine-tuning and inference-time control, which are arising as powerful tools to guide GMPDs towards desired regions of the learned protein distributions (e.g., those encoding target design properties), giving us tighter control over the protein design process.

However, a recurring limitation of both families is poor out-of-distribution (OOD) generalization. Inference-time control presupposes that GMPDs already encode rich structure over the relevant regions of sequence space; steering cannot along produce representations that the base model never learned. Fine tuning methods may allow for more flexibility, enabling to reshape the GMPD's learned distribution by changing its parameters. However, many curated datasets used for supervised fine-tuning and in-silico scoring methods used for RL are based on the natural distribution of proteins, reflecting the same biases and evolutionary constraints we would like to move away from [e.g., 69, 18, 60].

In this regard, including physics-based scoring methods, like RoseTTA fold [70] and FoldX [71], in the RL fine-tuning process may provide a promising avenue, enabling to move away from models that exclusively infer information from the natural distribution of proteins. [72].

Moreover, as GMPDs are pushed to explore protein regions distant from the natural distribution of proteins, experimental testing providing reliable validation is paramount. In this regard, lab-in-the-loop frameworks combining GMPDs with

experimental validation are a very promising avenue. Advances in laboratory automation [73] and techniques[74, 75] are reducing experimental bottlenecks, enabling faster experimental testing and unprecedented scale of screening.

A complementary strategy is to improve the data distribution seen during pre-training and downstream optimization: diversifying sequence sampling across the tree of life [76]. Other approaches introduce inductive biases, leveraging evolutionary context provided by Multiple Sequence Alignment to provide more efficient training [77] or sampling[22].

The field of generative modeling for protein research is advancing at an unprecedented pace. A multitude of techniques for the guidance and control of GMPDs has emerged in the past two years, demonstrating promise for the targeted engineering of proteins across diverse applications. As the field progresses, a central challenge will be extending these models beyond the natural sequence distribution—ensuring reliable performance in out-of-distribution (OOD) regimes and paving the way toward truly generalizable, controllable, and creative protein design.

1.4 Declaration of competing interest

The authors declare no conflict of interest.

1.5 Acknowledgments

the European Union nor the granting authority can be held responsible for them.

References

[1] W. Ahern, J. Yim, D. Tischer, S. Salike, S. M. Woodbury, D. Kim, I. Kalvet, Y. Kipnis, B. Coventry, H. R. Altae-Tran, M. Bauer, R. Barzilay, T. S. Jaakkola, R. Krishna, D. Baker, Atom level enzyme active site scaffolding using RFdiffusion2, bioRxivPublisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2025/04/10/2025.04.09.648075.full.pdf (2025). `doi:10.1101/2025.04.09.648075`.
URL `https://www.biorxiv.org/content/early/2025/04/10/2025.04.09.648075`

[2] J. B. Ingraham, M. Baranov, Z. Costello, K. W. Barber, W. Wang, A. Ismail, V. Frappier, D. M. Lord, C. Ng-Thow-Hing, E. R. Van Vlack, S. Tie, V. Xue, S. C. Cowles, A. Leung, J. V. Rodrigues, C. L. Morales-Perez, A. M. Ayoub, R. Green, K. Puentes, F. Oplinger, N. V. Panwar, F. Obermeyer, A. R. Root, A. L. Beam, F. J. Poelwijk, G. Grigoryan, Illuminating protein space with a programmable generative model, Nature 623 (7989) (2023) 1070–1078, publisher: Nature Publishing Group. `doi:10.1038/s41586-023-06728-8`.
URL `https://www.nature.com/articles/s41586-023-06728-8`

[3] H. Stark, F. Faltings, M. Choi, Y. Xie, E. Hur, T. O'Donnell, A. Bushuiev, T. Uçar, S. Passaro, W. Mao, M. Reveiz, R. Bushuiev, T. Pluskal, J. Sivic, K. Kreis, A. Vahdat, S. Ray, J. T. Goldstein, A. Savinov, J. A. Hambalek, A. Gupta, D. A. Taquiri-Diaz, Y. Zhang, A. K. Hatstat, A. Arada, N. H. Kim, E. Tackie-Yarboi, D. Boselli, L. Schnaider, C. C. Liu, G.-W. Li, D. Hnisz, D. M. Sabatini, W. F. DeGrado, J. Wohlwend, G. Corso, R. Barzilay, T. Jaakkola, Boltzgen: Toward universal binder design, PreprintAvailable at `https://github.com/HannesStark/boltzgen` (2025).

[4] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, Proceedings of the National Academy of Sciences 118 (15) (2021) e2016239118. `arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.2016239118`, `doi:10.1073/pnas.2016239118`.
URL `https://www.pnas.org/doi/abs/10.1073/pnas.2016239118`

[5] R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, A. Rives, Msa transformer, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, Vol. 139 of

Proceedings of Machine Learning Research, 2021, pp. 8844–8856.
URL https://proceedings.mlr.press/v139/rao21a.html

[6] N. Ferruz, S. Schmidt, B. Höcker, ProtGPT2 is a deep unsupervised language model for protein design, Nature Communications 13 (1) (2022) 4348. doi:10.1038/s41467-022-32007-7.
URL https://www.nature.com/articles/s41467-022-32007-7

[7] A. Bhatnagar, S. Jain, J. Beazer, S. C. Curran, A. M. Hoffnagle, K. Ching, M. Martyn, S. Nayfach, J. A. Ruffolo, A. Madani, Scaling unlocks broader generation and deeper functional understanding of proteins, bioRxiv (2025) 2025–04.

[8] E. Nguyen, M. Poli, M. G. Durrant, B. Kang, D. Katrekar, D. B. Li, L. J. Bartie, A. W. Thomas, S. H. King, G. Brixi, J. Sullivan, M. Y. Ng, A. Lewis, A. Lou, S. Ermon, S. A. Baccus, T. Hernandez-Boussard, C. Ré, P. D. Hsu, B. L. Hie, Sequence modeling and design from molecular to genome scale with evo, Science 386 (6723) (2024) eado9336. arXiv:https://www.science.org/doi/pdf/10.1126/science.ado9336, doi:10.1126/science.ado9336.
URL https://www.science.org/doi/abs/10.1126/science.ado9336

[9] G. Brixi, M. G. Durrant, J. Ku, M. Poli, G. Brockman, D. Chang, G. A. Gonzalez, S. H. King, D. B. Li, A. T. Merchant, et al., Genome modeling and design across all domains of life with evo 2, BioRxiv (2025) 2025–02.

[10] M. Pacesa, L. Nickel, C. Schellhaas, J. Schmidt, E. Pyatova, L. Kissling, P. Barendse, J. Choudhury, S. Kapoor, A. Alcaraz-Serna, Y. Cho, K. H. Ghamary, L. Vinué, B. J. Yachnin, A. M. Wollacott, S. Buckley, A. H. Westphal, S. Lindhoud, S. Georgeon, C. A. Goverde, G. N. Hatzopoulos, P. Gönczy, Y. D. Muller, G. Schwank, D. C. Swarts, A. J. Vecchio, B. L. Schneider, S. Ovchinnikov, B. E. Correia, One-shot design of functional protein binders with BindCraft, Nature (2025). doi:10.1038/s41586-025-09429-6.
URL https://www.nature.com/articles/s41586-025-09429-6

[11] R. Krishna, J. Wang, W. Ahern, P. Sturmfels, P. Venkatesh, I. Kalvet, G. R. Lee, F. S. Morey-Burrows, I. Anishchenko, I. R. Humphreys, R. McHugh, D. Vafeados, X. Li, G. A. Sutherland, A. Hitchcock, C. N. Hunter, A. Kang, E. Brackenbrough, A. K. Bera, M. Baek, F. DiMaio, D. Baker, Generalized biomolecular modeling and design with rosettafold all-atom, Science 384 (6693) (2024) eadl2528. arXiv:https://www.science.org/doi/pdf/10.1126/science.adl2528, doi:10.1126/science.adl2528.
URL https://www.science.org/doi/abs/10.1126/science.adl2528

[12] M. Braun, A. Tripp, M. Chakatok, S. Kaltenbrunner, M. Totaro, D. Stoll, A. Bijelic, W. Elaily, S. Y. Hoch, M. Aleotti, M. Hall, G. Oberdorfer, Computational design of highly active de novo enzymes, bioRxivPublisher: Cold Spring Harbor Laboratory _eprint:

https://www.biorxiv.org/content/early/2024/08/03/2024.08.02.606416.full.pdf (2024). `doi:10.1101/2024.08.02.606416`.
URL `https://www.biorxiv.org/content/early/2024/08/02.606416`

[13] K. K. Yang, S. Alamdari, A. Lee, K. Kaymak-Loveless, S. Char, G. Brixi, C. Domingo-Enrich, C. Wang, S. Lyu, N. Fusi, et al., The dayhoff atlas: scaling sequence diversity for improved protein generation, bioRxiv (2025) 2025–07.

[14] A. Cornman, J. West-Roberts, A. P. Camargo, S. Roux, M. Beracochea, M. Mirdita, S. Ovchinnikov, Y. Hwang, The omg dataset: An open metagenomic corpus for mixed-modality genomic language modeling, Cold Spring Harbor Laboratory (2024). `doi:10.1101/2024.08.14.607850`.
URL `https://www.biorxiv.org/content/early/2024/08/17/2024.08.14.607850`

[15] M. Varadi, D. Bertoni, P. Magana, U. Paramval, I. Pidruchna, M. Radhakrishnan, M. Tsenkov, S. Nair, M. Mirdita, J. Yeo, et al., Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences, Nucleic acids research 52 (D1) (2024) D368–D375.

[16] B. Chen, X. Cheng, P. Li, Y.-a. Geng, J. Gong, S. Li, Z. Bei, X. Tan, B. Wang, X. Zeng, et al., xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein, arXiv preprint arXiv:2401.06199 (2024).

[17] X. Cheng, B. Chen, P. Li, J. Gong, J. Tang, L. Song, Training compute-optimal protein language models, Advances in Neural Information Processing Systems 37 (2024) 69386–69418.

[18] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, et al., Simulating 500 million years of evolution with a language model, Science 387 (6736) (2025) 850–858.

[19] C. Gordon, A. X. Lu, P. Abbeel, Protein language model fitness is a matter of preference, bioRxiv (2024) 2024–10.

[20] F. Ding, J. Steinhardt, Protein language models are biased by unequal sequence sampling across the tree of life, BioRxiv (2024).

[21] P. Avasthi, R. York, The known protein universe is phylogenetically biased, Arcadia Science (aug 1 2024). `doi:10.57844/arcadia-570f-5cfb`.
URL `https://research.arcadiascience.com/pub/result-protein-universe-phylogenetic-bias/release/2`

[22] C. W. J. Pugh, P. G. Nuñez-Valencia, M. Dias, J. Frazer, From likelihood to fitness: Improving variant effect prediction in protein and genome language models (2025). `doi:10.1101/2025.05.20.655154`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.05.20.655154`

[23] C. Hou, D. Liu, A. Zafar, Y. Shen, Understanding language model scaling on protein fitness prediction, bioRxiv (2025).

[24] G. Munsamy, R. Illanes-Vicioso, S. Funcillo, I. T. Nakou, S. Lindner, G. Ayres, L. S. Sheehan, S. Moss, U. Eckhard, P. Lorenz, N. Ferruz, Conditional language models enable the efficient design of proficient enzymes (May 2024). `doi:10.1101/2024.05.03.592223`.
URL `http://biorxiv.org/lookup/doi/10.1101/2024.05.03.592223`

[25] A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos Jr, C. Xiong, Z. Z. Sun, et al., Large language models generate functional protein sequences across diverse families, Nature Biotechnology 41 (9) (2023) 1099–1106. `doi:10.1038/s41587-022-01618-2`.

[26] J. A. Ruffolo, S. Nayfach, J. Gallagher, A. Bhatnagar, J. Beazer, R. Hussain, J. Russ, J. Yip, E. Hill, M. Pacesa, A. J. Meeske, P. Cameron, A. Madani, Design of highly functional genome editors by modeling the universe of CRISPR-Cas sequences (Apr. 2024). `doi:10.1101/2024.04.22.590591`.
URL `http://biorxiv.org/lookup/doi/10.1101/2024.04.22.590591`

[27] D. Ivančić, A. Agudelo, J. Lindstrom-Vautrin, J. Jaraba-Wallace, M. Gallo, R. Das, A. Ragel, J. Herrero-Vicente, I. Higueras, F. Billeci, M. Sanvicente-García, P. Petazzi, N. Ferruz, A. Sánchez-Mejías, M. Güell, Discovery and protein language model-guided design of hyperactive transposases, Nature Biotechnology (Oct. 2025). `doi:10.1038/s41587-025-02816-4`.
URL `https://doi.org/10.1038/s41587-025-02816-4`

[28] S. H. King, C. L. Driscoll, D. B. Li, D. Guo, A. T. Merchant, G. Brixi, M. E. Wilkinson, B. L. Hie, Generative design of novel bacteriophages with genome language models, bioRxiv (2025) 2025–09.

[29] F. Stocco, M. Artigues-Lleixa, A. Hunklinger, T. Widatalla, M. Guell, N. Ferruz, Guiding generative protein language models with reinforcement learning (2025). `arXiv:2412.12979[q-bio]`, `doi:10.48550/arXiv.2412.12979`.
URL `http://arxiv.org/abs/2412.12979`

[30] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, Y. Gal, Ai models collapse when trained on recursively generated data, Nature 631 (8022) (2024) 755–759.

[31] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, et al., Deepseek-r1 incentivizes reasoning in llms through reinforcement learning, Nature 645 (8081) (2025) 633–638.

[32] M. Olivecrona, T. Blaschke, O. Engkvist, H. Chen, Molecular de-novo design through deep reinforcement learning, Journal of cheminformatics 9 (1) (2017) 48.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[34] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al., Deepseekmath: Pushing the limits of mathematical reasoning in open language models, arXiv preprint arXiv:2402.03300 (2024).

[35] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, C. Finn, Direct preference optimization: Your language model is secretly a reward model (2024). `arXiv:2305.18290[cs]`, `doi:10.48550/arXiv.2305.18290`.
URL `http://arxiv.org/abs/2305.18290`

[36] N. Blalock, S. Seshadri, A. Babbar, S. A. Fahlberg, A. Kulkarni, P. A. Romero, Functional alignment of protein language models via reinforcement learning (2025). `doi:10.1101/2025.05.02.651993`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.05.02.651993`

[37] F. Renard, C. Courtot, A. Reichlin, O. Bent, Model-based reinforcement learning for protein backbone design, arXiv:2405.01983 [cs] (May 2024). `doi:10.48550/arXiv.2405.01983`.
URL `http://arxiv.org/abs/2405.01983`

[38] T. Widatalla, R. Rafailov, B. Hie, Aligning protein generative models with experimental fitness via direct preference optimization (2024). `doi:10.1101/2024.05.20.595026`.
URL `http://biorxiv.org/lookup/doi/10.1101/2024.05.20.595026`

[39] H.-C. Gasser, D. A. Oyarzún, J. A. Alfaro, A. Rajan, Tuning ProteinMPNN to reduce protein visibility via MHC Class I through direct preference optimization, Protein Engineering, Design and Selection 38 (2025) gzaf003. `doi:10.1093/protein/gzaf003`.
URL `https://doi.org/10.1093/protein/gzaf003`

[40] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (3) (1992) 229–256.

[41] X. Xu, T. Xu, J. Zhou, X. Liao, R. Zhang, Y. Wang, L. Zhang, X. Gao, Ab-gen: antibody library design with generative pre-trained transformer and deep reinforcement learning, Genomics, Proteomics & Bioinformatics 21 (5) (2023) 1043–1053.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms (2017). `arXiv:1707.06347[cs]`, `doi:10.48550/arXiv.1707.06347`.
URL `http://arxiv.org/abs/1707.06347`

[43] Z. Wang, J. Fan, R. Guo, T. Nguyen, H. Ji, G. Liu, Proteinzero: Self-improving protein generation via online reinforcement learning, arXiv preprint arXiv:2506.07459 (2025).

[44] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science 362 (6419) (2018) 1140–1144.

[45] Y. Wang, H. Tang, L. Huang, L. Pan, L. Yang, H. Yang, F. Mu, M. Yang, Self-play reinforcement learning guides protein engineering, Nat Mach Intell 5 (8) (2023) 845–860. doi:10.1038/s42256-023-00691-9.
URL https://www.nature.com/articles/s42256-023-00691-9

[46] H. Lin, C. Zhu, T. Shang, N. Zhu, K. Lin, C. Zhang, X. Shao, X. Wang, H. Duan, Highplay: Cyclic peptide sequence design based on reinforcement learning and protein structure prediction, Journal of Medicinal Chemistry (2025).

[47] R. Park, D. J. Hsu, C. B. Roland, M. Korshunova, C. Tessler, S. Mannor, O. Viessmann, B. Trentini, Improving inverse folding for peptide design with diversity-regularized direct preference optimization, arXiv preprint arXiv:2410.19471 (2024).

[48] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Xu, H. Ding, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Chen, J. Yuan, J. Tu, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. You, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Zhou, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, Z. Zhang, DeepSeek-r1 incentivizes reasoning in LLMs through reinforcement learning, Nature 645 (8081) (2025) 633–638. doi:10.1038/s41586-025-09422-z.
URL https://www.nature.com/articles/s41586-025-09422-z

[49] H. Cao, H. Zhang, J. Xu, Z. Zhang, L. Shen, M. Sun, G. Liu, J. Xu, W.-J. Li, J. Ni, et al., From supervision to exploration: What does protein language model learn during reinforcement learning?, arXiv preprint arXiv:2510.01571 (2025).

[50] J. Yang, A. Bhatnagar, J. A. Ruffolo, A. Madani, Function-Guided Conditional Generation Using Protein Language Models with Adapters, arXiv:2410.03634 [q-bio] (Jun. 2025). `doi:10.48550/arXiv.2410.03634`.
URL `http://arxiv.org/abs/2410.03634`

[51] J. Dauparas, G. R. Lee, R. Pecoraro, L. An, I. Anishchenko, C. Glasscock, D. Baker, Atomic context-conditioned protein sequence design using LigandMPNN, Nat Methods 22 (4) (2025) 717–723. `doi:10.1038/s41592-025-02626-1`.
URL `https://www.nature.com/articles/s41592-025-02626-1`

[52] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, M. Steinegger, Colabfold: making protein folding accessible to all, Nature methods 19 (6) (2022) 679–682.

[53] S. Alamdari, N. Thakkar, R. Van Den Berg, N. Tenenholtz, R. Strome, A. M. Moses, A. X. Lu, N. Fusi, A. P. Amini, K. K. Yang, Protein generation with evolutionary diffusion: sequence is all you need (2023). `doi:10.1101/2023.09.11.556673`.
URL `http://biorxiv.org/lookup/doi/10.1101/2023.09.11.556673`

[54] Y. Cho, M. Pacesa, Z. Zhang, B. E. Correia, S. Ovchinnikov, Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design (2025). `doi:10.1101/2025.04.06.647261`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.04.06.647261`

[55] B. Zhou, L. Zheng, B. Wu, K. Yi, B. Zhong, Y. Tan, Q. Liu, P. Liò, L. Hong, A conditional protein diffusion model generates artificial programmable endonuclease sequences with enhanced activity, Cell Discovery 10 (1) (2024) 95.

[56] T. Truong Jr, T. Bepler, Poet: A generative model of protein families as sequences-of-sequences, Advances in Neural Information Processing Systems 36 (2023) 77379–77415.

[57] R. Weitzman, P. M. Groth, L. V. Niekerk, A. Otani, Y. Gal, D. Marks, P. Notin, Protriever: End-to-end differentiable protein homology search for fitness prediction (2025). `arXiv:2506.08954[q-bio]`, `doi:10.48550/arXiv.2506.08954`.
URL `http://arxiv.org/abs/2506.08954`

[58] T. F. Truong Jr, T. Bepler, Understanding protein function with a multimodal retrieval-augmented foundation model, arXiv preprint arXiv:2508.04724 (2025).

[59] B. L. Hie, V. R. Shanker, D. Xu, T. U. J. Bruun, P. A. Weidenbacher, S. Tang, W. Wu, J. E. Pak, P. S. Kim, Efficient evolution of human antibodies from general protein language models, Nature Biotechnology 42 (2) (2024) 275–283. doi:10.1038/s41587-023-01763-2.
URL https://doi.org/10.1038/s41587-023-01763-2

[60] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. De Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, D. Baker, Robust deep learning–based protein sequence design using ProteinMPNN, Science 378 (6615) (2022) 49–56. doi:10.1126/science.add2187.
URL https://www.science.org/doi/10.1126/science.add2187

[61] C. Zhou, Y. Qiu, T. Ling, J. Li, S. Liu, X. Wang, J. Song, W. Xiang, Cmadiff: Cross-modal aligned diffusion for controllable protein generation, arXiv preprint arXiv:2503.21450 (2025).

[62] E. Adams, L. Bai, M. Lee, Y. Yu, M. AlQuraishi, From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models, bioRxivPublisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2025/02/08/2025.02.06.636901.full.pdf (2025). doi:10.1101/2025.02.06.636901.

[63] E. N. V. Garcia, A. Ansuini, Interpreting and steering protein language models through sparse autoencoders, version Number: 1 (2025). doi:10.48550/ARXIV.2502.09135.
URL https://arxiv.org/abs/2502.09135

[64] N. Parsan, D. J. Yang, J. J. Yang, Towards Interpretable Protein Structure Prediction with Sparse Autoencoders, arXiv:2503.08764 [q-bio] (Mar. 2025). doi:10.48550/arXiv.2503.08764.
URL http://arxiv.org/abs/2503.08764

[65] G. B. Corominas, F. Stocco, N. Ferruz, Sparse autoencoders in protein engineering campaigns: Steering and model diffing, 2025.
URL https://openreview.net/forum?id=rnJ6Nn1Wf5

[66] J. Xiong, H. Nisonoff, M. Lukarska, I. Gaur, L. M. Oltrogge, D. F. Savage, J. Listgarten, Guide your favorite protein sequence generative model, arXiv preprint arXiv:2505.04823 (2025).

[67] N. Ferruz, S. Schmidt, B. Höcker, Protgpt2 is a deep unsupervised language model for protein design, Nature communications 13 (1) (2022) 4348.

[68] J. T. Darmawan, Y. Gal, P. Notin, Sampling protein language models for functional protein design, in: Learning Meaningful Representations of Life (LMRL) Workshop at ICLR 2025, 2025.

[69] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al., Evolutionary-scale prediction of atomic-level protein structure with a language model, Science 379 (6637) (2023) 1123–1130.

[70] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al., Accurate prediction of protein structures and interactions using a three-track neural network, Science 373 (6557) (2021) 871–876.

[71] J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, L. Serrano, The foldx web server: an online force field, Nucleic acids research 33 (suppl_2) (2005) W382–W388.

[72] F. Noé, S. Olsson, J. Köhler, H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science 365 (6457) (2019) eaaw1147, publisher: American Association for the Advancement of Science. `doi:10.1126/science.aaw1147`.
URL `https://www.science.org/doi/10.1126/science.aaw1147`

[73] J. T. Rapp, B. J. Bremer, P. A. Romero, Self-driving laboratories to autonomously navigate the protein fitness landscape, Nat Chem Eng 1 (1) (2024) 97–107. `doi:10.1038/s44286-023-00002-4`.
URL `https://www.nature.com/articles/s44286-023-00002-4`

[74] J. Qian, L. F. Milles, B. I. M. Wicky, A. Motmaen, X. Li, R. D. Kibler, L. Stewart, D. Baker, Accelerating protein design by scaling experimental characterization (2025). `doi:10.1101/2025.08.05.668824`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.08.05.668824`

[75] M. R. Stammnitz, B. Lehner, The genetic architecture of an allosteric hormone receptor, bioRxiv (2025) 2025–05.

[76] O. Vince, P. Oldach, V. Pereno, M. H. Y. Leung, C. Greco, G. Minto-Cowcher, S. Ur-Rehman, K. Y. K. Kam, W. Chow, E. Bolton, B. R. Mwambingu, N. L. Greenhalgh, I. E. Knot, L. Christoffersen, M. Clark, R. Pecoraro, A. W. Kollasch, T. Bohnuud, M. Bakalar, P. Lorenz, G. Gowers, Breaking through biology's data wall: Expanding the known tree of life by over 10x using a global biodiscovery pipeline (2025). `doi:10.1101/2025.06.11.658620`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.06.11.658620`

[77] Y. Akiyama, Z. Zhang, M. Mirdita, M. Steinegger, S. Ovchinnikov, Scaling down protein language modeling with MSA pairformer (2025). `doi:10.1101/2025.08.02.668173`.
URL `http://biorxiv.org/lookup/doi/10.1101/2025.08.02.668173`

Papers of particular interest, published within the period of review, have been highlighted as: * of special interest ** of outstanding interest.

2 Supplementary

2.1 Reinforcement Learning Losses (Supplementary to Table 2)

This section details the mathematical formulations of the reinforcement learning (RL) objectives summarized in Table 2. Here π_θ denotes the policy, while the $D_{KL}(\pi_\theta||\pi_{ref})$ is a KL regularization term to penalize strong deviations from the reference model π_{ref}.

REINFORCE. The REINFORCE algorithm maximizes the expected reward by increasing the log-likelihood of sampled actions by their associated return $r(x)$. The corresponding loss is:

$$\mathcal{L}_{\text{REINFORCE}} = -\mathbb{E}_{x \sim \pi_\theta} \left[r(x) \log \pi_\theta(x) \right]$$

Proximal Policy Optimization (PPO). PPO stabilizes training by preventing excessively large policy updates that may cause divergence. This is achieved by clipping the policy-ratio term $r_i(\theta) = \pi_\theta(a_i|s_i)/\pi_{\text{ref}}(a_i|s_i)$. The clipped objective is:

$$\mathcal{L}_{\text{PPO}} = -\frac{1}{G} \sum_{i=1}^{G} \left[\min \left(r_t(\theta) \, \hat{A}_t, \, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \, \hat{A}_t \right) \right]$$

where $\hat{A}_t$ denotes the advantage and ϵ defines the clipping range. While the advantage can be computed in different ways (e.g., GRPO-style), in PPO, it is classically computed relatived to a learned value function predicted the per-token expected reward. In RLHF, a KL penalty relative to a fixed reference model is added to the reward, ensuring that the new policy remains close to this reference.

AlphaZero. Trains a single neural network with two heads, one that predicts the probability distribution over actions $p_\theta(a|s)$ and the other that predicts the expected sum of rewards given a state s $v_\theta(s)$. The training data comes

from *self-play* using a Monte Carlo Tree Search (MCTS) that produces the final real outcome z and the used policies π (i.e visit counts distribution for MCTS rollouts). The training loss is the following:

$$\mathcal{L}(\theta) = (z - v_\theta(s))^2 \ - \sum_a \pi(a|s) \log p_\theta(a|s) \ + \ c \, \|\theta\|_2^2$$

where $(z - v_\theta(s))^2$ is the value loss (mean square error between predicted and true value), $-\sum_a \pi(a|s) \log p_\theta(a|s)$ is the policy loss (cross entropy between MCTS policy and the network policy and $c \, \|\theta\|_2^2$ is a regularization term to avoid large weights updates (scaled by a constat c).

Direct Preference Optimization (DPO). Introduced by Rafailov *et al.* (2023), DPO learns the optimal policy directly from pairwise preference data without requiring a reward model or value function. Given preferred (x_w) and dispreferred (x_l) samples, the objective is:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x_w, x_l)} \big[\log \sigma \big(\beta [\log \pi_\theta(x_w) - \log \pi_\theta(x_l)] \big) \big] \, ,$$

where σ is the sigmoid function and β controls the strength of preference separation.

Group Relative Policy Optimization (GRPO). GRPO extends PPO by organizing responses into $|o|$ groups and computing advantages relative to each group's mean performance, thereby reducing variance of each update and improving stability. It does not require a separate value model. The objective is:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \Big(r_{i,t}(\theta) \, \hat{A}_{i,t}, \, \text{clip}(r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \, \hat{A}_{i,t} \Big) + \beta \, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \, ,$$

where each group o_i contains trajectories with similar characteristics. $\hat{A}_{i,t}$ denotes the group-relative advantage, which is computed as,

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}_t(r)}{\text{std}_t(r)} \tag{1}$$

with the mean_t and std_t being computed over the group response.