PROTGPT3: AN OPEN-SOURCE FAMILY OF PROMPTABLE AND ALIGNED PROTEIN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative protein language models (pLMs) enable exploration of vast sequence spaces, but controlling generation toward desired functional families remains challenging, particularly in low-data regimes. We introduce ProtGPT3, an open family of autoregressive pLMs from 112M to 10B parameters, comprising single-sequence and MSA-promptable models and, to our knowledge, the largest publicly released pLM trained exclusively with an autoregressive objective. We use ProtGPT3 to compare supervised fine-tuning against few-shot prompting with homologous sequences. Across 25 held-out protein families, prompting outperforms fine-tuning in the low-data regime and matches it as training-set size grows. In an extreme low-data case study on defluorinase enzymes, we generate enzymes by fine-tuning or prompting ProtGPT3 models with seven enzymes only, producing four new defluorinase with activity values in line with the natural ones. We further investigate two complementary ways to control generation. First, we apply direct preference optimization to align the models at a large scale, mitigating the well-established pretraining bias toward low-complexity sequences, while preserving sequence diversity. Second, we introduce a novel Feynman–Kac sequential Monte Carlo procedure that uses inference-time compute to steer generation toward sequence-level objectives without updating model parameters. We release model weights, configurations, training and inference code on Hugging Face (https://huggingface.co/protgpt3) under a pseudonym.

1 INTRODUCTION

Proteins found in nature represent a vanishingly small sample of the astronomical number of possible sequences that can be formed. Natural proteins perform a remarkable diversity of functions, yet their properties are shaped by evolutionary pressures that rarely align with industrial or therapeutic requirements. The design of artificial proteins with novel and customizable functions has therefore been a central goal of biochemistry. Protein Language Models (pLMs) provide a means of learning from natural sequence diversity and generating novel proteins (Lin et al., 2023; Verkuil et al., 2022). Autoregressive pLMs are particularly well-suited to generation and have shown strong generative capabilities, enabling the experimental design of several functional proteins, often with sequence identities as low as 30% to known natural proteins (Middendorf & Ferruz, 2026). However, the engineering of proteins not only require the generation of plausible sequences; but controlling generation toward desired properties.

While the development of pLMs has mirrored advances in NLP, several approaches routinely used for the controllability of large language models (LLMs) remain underexplored in protein engineering. First, modern LLMs are routinely post-trained to *align* them with user intent, with the goal of removing inherent pre-training biases and make them more useful. This primarily relies on reinforcement learning (RL) and preference-optimization methods (Rafailov et al., 2023). Although RL is emerging as a strategy for aligning generative pLMs for protein design (Stocco et al., 2026), studies of pLM alignment at large scale remain limited. Second, following GPT-2, LLMs increasingly shifted from task-specific fine-tuning toward few-shot prompting (Radford et al., 2019; Brown et al., 2020), in which a model is conditioned (prompted) on a small number of examples at inference time without updating its weights. In protein engineering, however, users still largely rely on fine-tuning comparatively small pLMs on task-specific datasets, with no clear comparison to few-shot prompting. Third, LLMs increasingly improve their outputs by allocating additional compute at inference time

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

rather than by updating weights. In protein design, inference-time steering has so far been developed mainly for diffusion-based structure generators, while autoregressive pLMs are typically sampled once per design. MSA-promptable models are a natural setting for this strategy: generation operates over sets of homologous sequences, enabling the scoring of full generated trajectories that can be further selected for re-prompting, ultimately directing toward an engineering objective. In this respect, we make the following contributions:

1) We release ProtGPT3 on HuggingFace, an open-source family of auto-regressive pLMs spanning 112M to 10B parameters, together with ProtGPT3-MSA, a 112M-parameter model for homolog-conditioned generation. Our 10B model is the largest publicly available auto-regressive pLM.

2) We test SFT vs few-shot homolog prompting for targeted protein generation across data regimes and model scaling, showing few-shot prompting is especially good in low data regimes. We include experimental validation, generating novel active defluorinase enzymes in an extremely data-scarce setting, increasing the number of known defluorinases from seven to eleven active sequences.

3) We introduce two steering methods for our models: a novel FK-SMC procedure that improves sequence stability and fitness at inference time, and large-scale post-training alignment that reduces pretraining biases towards low-complexity generation while preserving diversity.

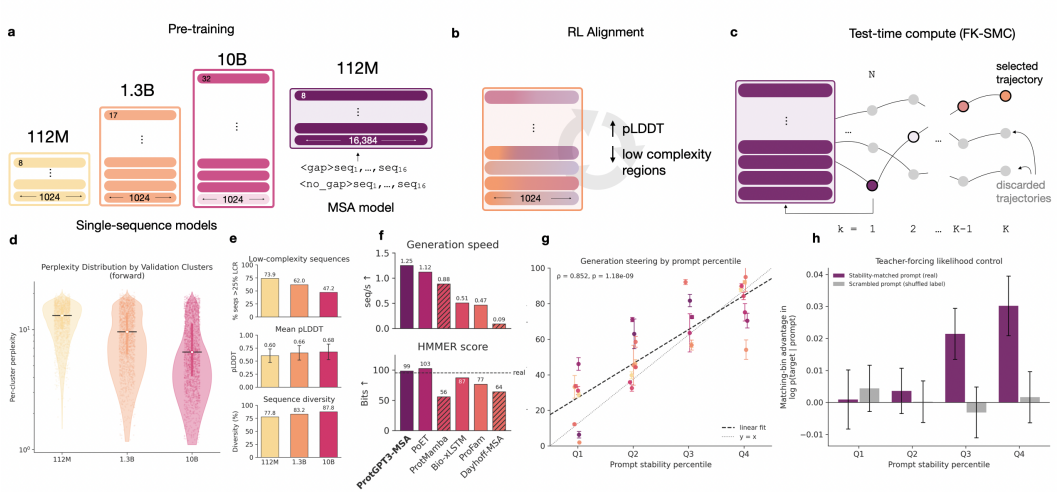

Figure 1: **Paper overview** (top row): **(a)** Pre-training of 3 single-sequences auto-regressive pLMs (from 112M to 10B parameters) and one MSA pLM, which can be few-short prompted with up to 15 aligned or unaligned homologous sequences. **(b)** Large scale (RL) alignment of 3 single sequence models to improve sequence quality. **(c)** Development of a novel test-time compute strategy (FK-SMC) to steer MSA models generation towards a target property. **Models' performance** (bottom row): **(d)** Larger models achieve lower perplexity across the full distribution of held-out validation protein clusters. **(e)** Generation performance of single-sequence models improves with parameter scale across both quality and diversity metrics. **(f)** Compared to other MSA models, ProtGPT3-MSA is the fastest generator while closely matching the best-performing HMMER model. Top: generation speed (sequences/s). Bottom: mean HMMER bit score of generated sequences against the query-family profile. The dashed line shows performance for held-out real MSA sequences. **(g)** Few-shot prompting steers ProtGPT3-MSA generation towards variants with a target stability range, based on Megascale (Tsuboyama et al., 2023). Prompt stability percentile (x-axis, oracle-predicted quartile of the 15 prompt sequences) vs. the oracle-score percentile of the resulting generated sequences (y-axis); colored by tested wild-type (WT). Dashed line: linear fit (Spearman $\rho = 0.85$, $p < 0.001$); dotted line: $y = x$. (h) Ground-truth teacher-forcing control: for each prompted stability percentile, ProtGPT3-MSA assigns higher log-likelihood to (*real*) ground-truth (held-out) variants matching the prompt stability percentile range than ground-truth variants in a different stability percentile (for the top two percentiles, Q3 & Q4). This no longer holds when stability scores are randomly shuffled (*shuffled label* – negative control).Error bars: t-based 95% CI.

2 Related Work

Scaling and open source. Scaling model and dataset size has been central to autoregressive pLMs. RITA released a Hugging Face family of causal pLMs up to 1.2B parameters (Hesslow et al., 2022), while ProGen2 released checkpoints up to 6.4B (Nijkamp et al., 2023). Subsequent studies characterized compute-optimal training for causal and masked pLMs across model and data scales (Cheng et al., 2024; Serrano et al., 2024), with dedicated causal checkpoints released up to 7B parameters. More recent studies have reported substantially larger models, although the largest models trained are not always released. ProGen3 scaled sparse generative pLMs to 46B parameters, but its publicly released checkpoints currently stop at 3B (Bhatnagar et al., 2025). xTrimoPGLM released a quantized 100B checkpoint that supports causal generation, but it was pretrained with a unified masked and autoregressive objective (Chen et al., 2025). The Dayhoff Atlas released protein-only autoregressive models at 170M and 3B parameters, including models supporting homolog-conditioned generation (Yang et al., 2025). At the time of submission, ProtGPT3-10B is the largest publicly released checkpoint pretrained exclusively through auto-regressive next-amino-acid prediction. For ease of use, we release the complete ProtGPT3 family in standard Hugging Face Transformers format.

Alignment. Scaling pLMs improves representations and generation quality, but does not inherently align models with protein engineering objectives such as activity, binding affinity, or stability (Hoffmann et al., 2022; Cheng et al., 2024; Chen et al., 2024; Bhatnagar et al., 2025; Stocco et al., 2026). We use RL-alignment to describe parameter-updating methods that steer pretrained generative priors toward task objectives using feedback, in contrast to parameter-free inference-time steering (Stocco et al., 2026). Prior work has therefore focused on aligning pLMs with specific objectives: ESM-IF1 is aligned with stability using DPO (Widatalla et al., 2024), ProGen3 is aligned with ProteinGym and Megascale via IRPO (Bhatnagar et al., 2025), and PPO- or DPO-based approaches have been used for fluorescence optimization and EGFR binder engineering using experimental feedback (Blalock et al., 2025; Stocco et al., 2024). In contrast, we align our models to directly produce sequences with predicted structural order and minimal low complexity regions, so that they are directly aligned toward protein engineering, reducing the burden of post-generation filtering on the user(Bhatnagar et al., 2025) .

Few-shot family-conditioned generation. Recent protein generative models condition on homologous sequences as in-context examples of a target family, analogous to prompting in LLMs. Early approaches relied on aligned MSAs: MSA Transformer introduced row–column attention over homolog alignments (Rao et al., 2021), later enabling enzyme design applications (Johnson et al., 2025). More recent methods use unaligned homolog sets: PoET represents families as order-invariant sequence sets for retrieval-augmented generation (Truong Jr & Bepler, 2023), while ProtMamba and Bio-xLSTM use long-context sequence models for homology-aware generation (Sgarbossa et al., 2025; Schmidinger et al., 2024). ProFam evaluates whether generated families preserve natural conservation and covariance statistics suitable for AlphaFold2 inputs (Wells et al., 2025), and Dayhoff unifies single-sequence, aligned-, and unaligned-homolog conditioning, reporting experimental expression rates of 35% (Yang et al., 2025). While these works demonstrate the potential of prompting pLMs with homologous sequences for target family generation, a systematic comparison between this approach and the more traditional paradigm of supervised fine-tuning remains lacking.

3 Large-Scale Pretraining of a Family of Single-Sequence pLMs

3.1 Model architectures

ProtGPT3 is a family of autoregressive decoder-only pLMs based on a Mixtral-style sparse mixture-of-experts (MoE) architecture with 8 experts (Jiang et al., 2024). Compared to a dense Llama3 architecture (Grattafiori et al., 2024), we find Mixtral yields marginally worse validation perplexity but $\approx 1/3$ faster training and $\approx 1/2$ faster inference (Appendix A.1), consistent with prior pLM (Bhatnagar et al., 2025) and NLP work (Liu et al., 2024; Jiang et al., 2024). We trained three single-sequence models (ProtGPT3-112M, -1.3B and -10B) and one multiple sequence alignment (MSA) model, ProtGPT3-MSA (112M). All models are trained for causal language modeling in both directions (N-to-C and C-to-N), with the direction given by a special token at the start of each sequence (Appendix A.2). ProtGPT3-MSA starts from the final ProtGPT3-112M checkpoint and was trained to utoregressively predicts set of 16 concatenated sequences from the same MSA (up to

16,384 tokens), each separated by a special token. The sequences are passed to the model in either *aligned* (with gap tokens) or *unaligned* (full sequences, including insertions) mode, with a special token placed at the start of the 16 sequence specifying which modality (Appendix A.4).

3.2 DATASETS

We trained all single-sequence models on UniRef90 (Ahmad et al., 2025) and on non-singleton clusters of the GigaRef subset (1.89B sequences in 243M clusters at 50% identity) of the Dayhoff Atlas (Yang et al., 2025), which at the time of training provided the largest curated public protein dataset (3.34 billion sequences). Following Bhatnagar et al. (2025)'s approach for best out-of-distribution validation performance, we sample training sequences in two steps: we first draw a cluster c with probability $p(c) \propto \frac{n_c}{1+\log n_c}$, where n_c is the number of sequences in c, and then draw a sequence from c uniformly at random. UniRef sequences are critical for pLM training performance (Cheng et al., 2024; Yang et al., 2025; Meier et al., 2021). However, UniRef-derived clusters are under-represented in GigaRef, making up only 13M of GigaRef's 243M culsters. To overcome this and determine how best to combine the diversity of GigaRef with the quality of UniRef90, we trained the 112M-parameter model under four data regimes: (1) UniRef90 only, (2) GigaRef only, (3) mixed batches drawn 50/50 from UniRef90 and GigaRef, and (4) GigaRef alone, with a loss that upweights sequences derived from UniRef. Mixed 50/50 batches gave the best trade-off between UniRef90 and GigaRef validation perplexity on validations set with at most 50% sequence identity to training sequences (Appendix A.3, Fig. 5). We therefore used this data regime to train ProtGPT3 -112M and -1.3B. For ProtGPT3-10B, we used a 30/70 (Uniref90/Gigaref) mix since UniRef90 did not have enough tokens (sequences) to cover 50% of the optimal-compute budget for this model size. All single-sequence models were trained based on compute-optimal scaling laws for sparse pLMs (Bhatnagar et al., 2025; Cheng et al., 2024) (Appendix A.2). Finally, ProtGPT3-MSA was trained on 8.5M MSAs from OpenProteinSet Uniclust30 (Ahdritz et al., 2023) (Appendix A.4).

3.3 UNCONDITIONAL PROTEIN GENERATION ACROSS SCALE

To evaluate the sequence-only models across scale (112M upto 10B), we computed each model's perplexity on held-out validation sequences with at most 50% identity to its training set. We did this at the protein-cluster level rather than as a single average, capturing how uniformly each model covers the held-out sequence space. Larger models achieved lower perplexity not only on average but across the full distribution of clusters, progressively shrinking the tail of high-perplexity sequences (Fig 1c & Appendix A.5). We then assessed generation for sequence quality and diversity, generating 100 sequences per model via top-p sampling (Holtzman et al., 2019) for every combination of top-p $\{0.6, 0.8, 0.9, 1.0\}$ and temperature $\{0.6, 0.8, 0.9, 1.0\}$ across the 3 model sizes. Quality was measured by ESMFold-predicted pLDDT (Lin et al., 2023), the proportion of terminating sequences (correct EOS token), and the proportion of low-complexity proteins (>25% low-complexity regions) (Frith, 2011); diversity was measured by clustering generated sequences at 50% identity (e.g., Cheng et al., 2024). The largest model (ProtGPT3-10B) produced the highest-quality and most diverse sequences (Fig 1e). These results are aggregated across all top-p/temperature combinations, which masks a strong dependence on hyperparameters—specific configurations can yield substantially higher quality than the reported averages. Finally, we benchmarked all three models on ProteinGym (Notin et al., 2023), where the largest model again performed best (Table 2, Appendix). While our models perform comparably to other single-sequence autoregressive pLMs on substitution DMS datasets, ProtGPT3-10B ranks among the top 4 on indel DMS datasets.

3.4 CONDITIONAL PROTEIN GENERATION

Comparison to previous models ProtGPT3-MSA enables conditional generation by prompting with sets of up to 15 homologous sequences. We evaluate it against two conditioning strategies: a comprehensive set of prior MSA models (PoET, ProtMamba, Bio-xLSTM, ProFam, Dayhoff-MSA) and label-conditioned pLMs (Progen-1, ZymCTRL). For the first, we randomly sampled 25 MSAs from ProtGPT3-MSA's validation set and tested each model's ability to generate sequences consistent with the family statistics (per-column KL divergence, HMMER score) when prompted with the same 15 homologs and generating equal numbers of sequences per condition. ProtGPT3-MSA achieves the fastest generation while performing very close to the top model, PoET (Fig 1f & Table 3). Note that

this is a conservative test for ProtGPT3-MSA, since these MSAs are held out from its training set yet may fall within the other models' training data. For the label-conditioned comparison, we generate enzymes from 15 randomly sampled enzymatic families and score family membership with CLEAN (Yu et al., 2023). Since ZymCTRL and Progen-1 use different conditioning labels (EC numbers and UniProt keywords, respectively), they were benchmarked on a separate set of 15 MSAs. ProtGPT3-MSA greatly outperforms both abel-conditioned models (Tables 4 and 5). Note, we selected Progen-1 as a comparison, since later Progen models no longer implement label conditioning. Similarly to previous work (Wells et al., 2025; Truong Jr & Bepler, 2023), we find training on homologous sequences improves predictive power over single-sequence models on ProteinGym (Notin et al., 2023), despite a far smaller parameter count (112M vs. 10B; Table 2, Appendix). Finally, enriching the input with additional homologs via an ensemble strategy improves performance further, showing that evolutionary context enhances fitness estimation at a fixed parameter budget and yielding the best model in the ProtGPT family for this task (Appendix A.6).

Few-shot prompting enables stability-aware protein generation Next, we ask whether few-shot prompting can steer ProtGPT3-MSA generations toward sequences at a target stability range. To test this, we prompt ProtGPT3-MSA with 15-sequence prompt sets drawn from four ($\Delta\Delta G$) stability-percentile groups (Q1-Q4, lowest to highest) across ten MegaScale DMS libraries (Tsuboyama et al., 2023). We then ask whether the (predicted) stability of ProtGPT3-MSA generated sequences relates to the stability percentile of the sequences in the prompt. To do so, we train a separate oracle, ProtGPT3-112M, on the entire Megascale dataset (Tsuboyama et al., 2023), to rank the stability ($\Delta\Delta G$) percentile of ProtGPT3-MSA generated sequences (see Appendix A.8). Fig 1g show that the the stability of ProtGPT3-MSA generated sequences strongly correlates with the stability of the sequences used as prompt (spearman correlation of 0.85 and $p < 0.001$; panel a). Interestingly, for the first 3 percentiles (Q1-Q3), ProtGPT3-MSA has a tendency towards stabilizing the prompt sequences, generating sequence higher in stability than the prompt (i.e., above the gray dotted line in Fig 1g), while saturating for the highest percentile group, Q4. We further validate this effect on ground-truth data: ProtGPT3-MSA assigns higher likelihood to held-out Megascale sequences matching the prompt's ground-truth stability percentile than to sequences from other percentiles, an effect that disappears under a scrambled-label negative control (Fig 1h and Appendix A.9 for experimental details). In Appendix A.9, we show this effects is largely explained by the position of the mutations in the prompt sequences (Figure 8).

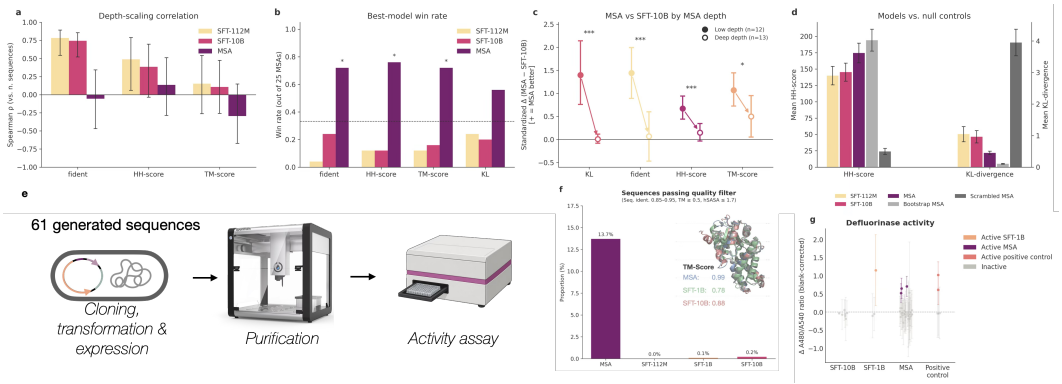

Figure 2: **ProtGPT3-MSA vs SFT across data and model scale.** **(a)** Spearman correlation between training-MSA depth and each performance metric across 3 models. **(b)** Fraction of the 25 validation MSAs on which each model achieves the best score (win rate); asterisks mark significance (chance $= 1/2$) for the winning model against the other two combined. **(c)** Mean standardized score difference between ProtGPT3-MSA and SFT-10B (positive = ProtGPT3-MSA better), separately for low-depth (< 340 Neff training sequences, $n = 12$, filled) and high-depth (≥ 340, $n = 13$, hollow) MSAs (see also Fig. 9 in the Appendix for correlation values); asterisks mark the depth-regime interaction test (Table 7). **(d)** Mean HH-score and KL-divergence for all three models alongside two controls: Bootstrap-MSA (upper-bound control) and Scrambled-MSA (a lower-bound control). **(e)** Experimental protocol. **(f)** Computational success rates of designed defluorinases generated with ProtGPT3-MSA and SFT models. **(g)** Experimental success of generated sequences and positive controls. Bars represent 95% bootstrap CIs.

4 FEW-SHOT PROMPTING VS SFT FOR CONDITIONAL PROTEIN GENERATION

We investigate whether ProtGPT3-MSA enables conditional protein generation without task-specific fine-tuning. Single-sequence autoregressive models are typically fine-tuned (SFT) on a target family to generate family-consistent sequences. By contrast, MSA models can be few-shot prompted with homologous sequences to induce conditional generation, without any weight updates. Yet a systematic, practical comparison of the two approaches, across data regimes and model scales, is lacking. To address this gap, we design a multi-step evaluation pipeline comparing (i) SFT single-sequence ProtGPT3 models and (ii) few-shot prompted ProtGPT3-MSA for conditional protein generation across diverse protein families. We compare these two approaches across two axes: data regime and model scale. For data regime, we evaluate generations 1) across 25 MSAs from ProtGPT3-MSA held-out validation set, spanning a broad range of depths (80-1750, number of effective sequences, Neff, (Morcos et al., 2011)) and, 2) an extremely data scarce defluorinases case study, including experimental validation with activity assay. For model scale, we compare ProtGPT3-MSA against SFT at both ends of the single-sequence model range (i.e, ProtGPT3-112M and ProtGPT3-10B). Importantly, all ProtGPT3-MSA prompts are built in unaligned mode for a fair comparison with the single-sequence models, which were not trained on aligned inputs (note, aligned mode performs comparably; Appendix Fig 11). Additionally, we employ a strict SFT pipeline to rule out protocol effects (see next section and Appendix A.10.1).

Conditional generation comparison for 25 held-out validation MSAs We first evaluate SFT versus few-shot prompting ProtGPT3-MSA across 25 MSAs. We sample these from ProtGPT3-MSA held-out validation set, using a stratified, log-spaced scheme to ensure coverage across MSA depths and a range of sequence ($20\% - 80\%$) and structural ($20\% - 80\%$) diversity (Table 8 and Appendix A.11 for sampling procedure). Each MSA is treated as an independent evaluation instance. For each sampled MSA, we construct random train/validation split, and use the training split to 1) fine-tune the two single-sequence ProtGPT3 models (112M and 10B) and 2) few-shot prompt ProtGPT3-MSA by randomly sub-sampling a set of 15 sequences. To ensure proper SFT, uncontaminated by protocol artifacts, we rely on a strict SFT pipeline: we first run a per-model-size learning-rate search on a random held-out MSA, followed by per-family SFT with early stopping to prevent overfitting. Appendix A.10.1 reports mean best train/validation loss and selected models' training epochs across all 25 MSAs, confirming that no SFT model was under- or over- trained. We then generated 90 sequences per model (10 sequences for each combination of top-p $\in \{0.7, 0.9, 1.0\}$ and temperature $\in \{0.8, 0.9, 1.0\}$). Finally, for each MSA, we compare the generated sequences from the 3 models against the corresponding held-out validation split (i.e., sequences not used for either SFT or prompting) using metrics of sequence-, structure-, and family-level consistency (Appendix A.10).

Fig. 2a shows SFT-112M and SFT-10B both improve with deeper MSAs on sequence- (fident) and family-level (HH-score) metrics, but not on structure (TM-score). This suggests more training data sharpens sequence/profile recovery, but not global fold, which both models broadly capture regardless of depth. For each MSA, both fident and TM-score are normalized by the maximum fident and maximum TM-score observed in the corresponding real train-to-validation splits, since the max operation naturally scales with the number of elements. ProtGPT3-MSA shows no MSA depth dependence on any metric. This is expected since the model is always prompted with 15 (random) sequences irrespective of the MSA depth. The more striking result is that ProtGPT3-MSA significantly outperforms both SFT models on the majority of the 25 validation MSAs, across sequence-, structure-, and family-level metrics alike; KL-divergence is the only metric where its win rate falls just short of significance (Fig. 2b). However, Fig. 2c show that this advantage of ProtGPT3-MSA over SFT is limited to shallower MSAs and vanishes when SFT has access to deeper families. We confirm this data-regime dependence, by showing a significant interaction effect between the ProtGPT3-MSA–SFT performance gap and the number of training sequences across all 4 metrics (see Table 7 in Appendix for p-values). For our analysis, we defined shallow and deep MSAs using a threshold of 340 Neff sequences (Morcos et al., 2011), chosen because it yields a balanced split across the two groups (12 shallow and 13 deep MSAs). This value should not be interpreted as a hard threshold for few-shot prompting vs SFT, as the exact cutoff likely depends on each specific scenario. To overcome this, Fig. 9 (Appendix) reports the correlation between ProtGPT3-MSA's performance advantage and the Neff of each MSA, showing that the performance gap shrinks with MSA depth independently of a threshold. Together, these results show that few-shot prompting delivers its greatest value precisely in the low-data regime, where fine-tuning is least practical. That advantage fades as training data accumulates, with SFT eventually catching up.

Additionally, Fig. 2b and Fig 10b (Appendix) show little performance difference between SFT-112M and SFT-10B performances, indicating no clear advantage of model scaling for SFT aimed at target protein family generation.

Finally, we include two baselines to get a concrete measure of how well ProtGPT3-MSA performs on family-level consistency metrics (i.e., HHalign and positional KL-divergence): "Bootstrap MSA" and "Scrambled MSA," which respectively approximate upper- and lower-bound performance on these metrics. Specifically, Bootstrap MSA samples 30% of each true MSA sequences, performing the analysis for sequences that come from the true family distribution for each MSA. Conversely, "Scrambled MSA" provides a negative control, by running the family-level consistency analysis for ProtGPT3-MSA generated sequences for each MSA against a different MSA. This ensures the generated sequences by ProtGPT3-MSA do not just match per-column statistics of any MSA. Fig 2d shows ProtGPT3-MSA generated sequences achieve family-level consistency metrics close to sequences truly sampled from those family across most tested MSAs (i.e., Bootstrap MSA), while scrambled, mismatched-family control MSAs performed very poorly confirming their generated sequences carry genuine family-specific signal. See Appendix A.10.6 for more details on the statistical tests used and their exact values.

Conditional generation comparison for Low-N design of defluorinase enzymes

Defluorinases were selected as a test case for experimental characterization under extreme data scarcity. These enzymes cleave carbon–fluorine bonds found in persistent pollutants such as PFAS. Defluorinases occur in the (S)-2-haloacid dehalogenase and α/β-hydrolase superfamilies, where defluorinating activity evolved independently (Wackett, 2025). Each superfamily has less than 10 experimentally annotated defluorinases, that are also highly similar to dechlorinases - enzymes cleaving carbon-chloride bonds. Because defluorinases are highly similar to dechlorinases and their distinguishing molecular features remain unclear (Probst et al., 2025), family-level datasets remain too ambiguous, whereas experimentally confirmed defluorinases are too scarce. Therefore, we set out to compare the generation of defluorinases from the (S)-2-haloacid dehalogenase superfamily via supervised fine-tuning of the single-sequence models and few-shot prompting of ProtGPT3-MSA with the 7 defluorinases identified and characterized by Probst et al. (2025). Again, we rely on a strict SFT pipeline with k-fold cross validation and early stopping to get the best SFT performance (Appendix A.14). The two approaches were evaluated first computationally, using a set of well-established filters (e.g., Verkuil et al., 2022; Xu & Zhang, 2010) (see Appendix A.14 for details), and then experimentally, for the surviving high-quality sequences. Few-shot generation with ProtGPT3-MSA achieved 148-fold and 63-fold higher computational success rates than fine-tuned ProtGPT3-1.3B and ProtGPT3-10B, respectively, while ProtGPT3-112M did not generate any successful sequence (Fig. 2g). ProtGPT3-MSA performed consistently better on average than the fine-tuned single sequence models, which produced successful sequences only as outliers from the distribution of their generations (Fig 13 in Appendix A.14). This agrees with our results on the 25 held-out MSAs, where few-shot prompting was most effective in low-data regimes. In Appendix A.14.1, we explore SFT failure on 7 training sequences only, including the impact of the chosen filters. For experimental validation, we selected all sequences passing the filters for the single-sequence models, providing 3 and 7 sequences for the 1.3B and 10B models. As ProtGPT3-MSA had many more sequences that passed the filters than what we could test experimentally, we randomly chose 51 filter-passing sequences for experimental testing (Appendix A.14). After cloning and purification, all candidates coming from ProtGPT3-1.3B and ProtGPT3-10B showed detectable amounts of protein by Bradford assay, while only 9 out of the 51 ProtGPT3-MSA designs failed in expression or purification. Finally, activity assays with the purified enzymes revealed 3 active candidates from ProtGPT3-MSA and 1 active candidate from ProtGPT3-1.3B, achieving activity yields comparable to those of WT sequences Q0SK70 and WP_087189991.1 (Fig 2g). The generated active sequences are reported in Table 12, increasing the number of known active defluorinase enzymes from seven to eleven active sequences.

5 Inference-time compute strategies in promptable pLMs

Autoregressive MSA pLMs like ProtGPT3-MSA are a natural fit for inference-time compute strategies, unlike single-sequence models. Because generation operates over sets of homologous sequences, the model yields a structured, sequence-level search space well suited to trajectory-based selection. This matters because many protein design objectives are defined at the full-sequence level, so scoring

functions can be built directly into homolog sampling. To exploit this, we develop a Feynman-Kac-style sequential Monte Carlo (FK-SMC) procedure that samples target-objective sequences from ProtGPT3-MSA (Algorithm 1, Appendix). Given a prompt of one or more concatenated homologs, ProtGPT3-MSA branches N candidate trajectories, each a K-step autoregressive rollout over homolog sequences. We score each trajectory across its full K-step continuation, keep the first homolog from the highest-scoring trajectory, append it to the prompt, and repeat until the 16th concatenated sequence is generated (i.e., ProtGPT3-MSA max number of sequences).

We test FK-SMC with ProtGPT3-MSA to increase the predicted stability of generate sequences, starting from low stability sequences as prompt, across 100 randomly sampled MegaScale DMS libraries (Tsuboyama et al., 2023). For each DMS, we randomly sample 6 variants from the lowest $\Delta\Delta G$ percentile (i.e., <25%) to be used as the starting prompt for FK-SMC. We then employ FK-SMC with the previously trained MegaScale stability oracle as scoring function (Section 3.4 Appendix A.8) to gradually steer ProtGPT3-MSA generation towards high stability sequences over 10 generation steps. Fig 3a show that FK-SMC increases sequence stability at each generation step, for both single- and double- mutations, confirming the procedure reliably steers generation towards the objective. Next, we compare FK-SMC with a matched-compute best-of-N baseline (i.e., same number of prompted, generated and scored sequences), across both single- and double- point mutations from MegaScale and going up to quadruple point mutations by training a separate oracle on the GB1 DMS dataset (Wu et al., 2016) (see Appendix A.15.1). Crucially, Fig 3c show that for single-point mutations, best-of-N and FK-SMC perform equivalently, both saturating performance by generating variants in top (predicted) stability percentiles (relative to the entire dataset). This is not unexpected, the space of single point mutation is limited, so extensive best-of-N is likely to land on at least one highly stable sequence matching FK-SMC performance. This is also evident in Fig 3a, where FK-SMC is close to performance saturation after the first step, which is equivalent to a best-of-N approach. However, for the larger space of double- and quadruple- point mutations, Fig 3c-d show that FK-SMC achieves higher performance across the compute spectrum, by efficiently leveraging the generated sequences as a self-steering prompt. Finally, we validate FK-SMC on ground truth stability data. Fig 3b shows that the prompt generated by FK-SMC make ProtGPT3-MSA more likely to generate ground truth Megascale variants in the top 90% stability percentile, while making it less likely to generate ground truth variants in the lowest 10% stability percentile.

6 Aligning single-sequence pLMs for improved sequence generation quality

A common LLM failure mode is repetitive generation, which in single-sequence pLMs manifests as low-complexity sequences (Welleck et al., 2019; Holtzman et al., 2019; Bhatnagar et al., 2025; Cheng et al., 2024). Although scaling partially alleviates this issue (Fig. 1e), all models still generate an excessive fraction of such sequences. To reduce this bias, we implement a large-scale reinforcement-learning alignment pipeline for each single-sequence model. We first generate 80,000 sequences per model across combinations of top-$p \in \{0.6, 0.8, 0.9, 1.0\}$ and temperature $\in \{0.6, 0.8, 0.9, 1.0\}$, scoring each by its fraction of low-complexity regions and its ESMFold-predicted pLDDT (Lin et al., 2023). Sequences with pLDDT > 0.7 and fewer than 25% low-complexity residues are treated as positives, all others as negatives; the pLDDT criterion acts as a proxy for structural plausibility, preventing trivial optimization toward random high-complexity strings. We then apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to bias generation toward positives and away from negatives, repeating the cycle for up to three iterations (Appendix A.16). Comparing pre- and post-alignment rejection sampling across all three models, alignment substantially improves generation quality: the proportion of high-complexity sequences rises while pLDDT stays high (Fig. 3e; Fig. 17b). The fraction of low-complexity generations drops by more than 20% for the 112M and 1.3B models and by nearly 50% for the 10B model. Diversity also increases across all scales (Fig. 3f), while pretraining validation perplexity is largely unchanged (Fig. 17c), indicating minimal catastrophic forgetting. Notably, a single iteration reduces low-complexity generations below 5% for ProtGPT3-10B, suggesting larger models are particularly responsive to alignment and easier to steer toward biologically realistic distributions (e.g., Bhatnagar et al., 2025). We publicly release the 3 DPO aligned model alongside the 4 base models.

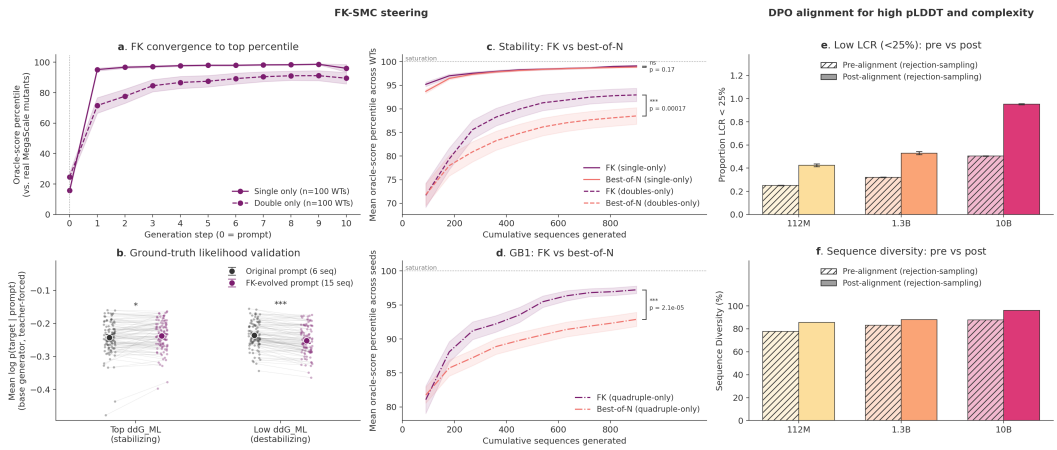

Figure 3: **Test-time steering with FK-SMC, and DPO alignment of base models.** Oracle scores are expressed as percentiles against real variants of the same mutation order; bands are 95% CI. **(a)** FK converges to generating top stability percentile sequences, starting from a deliberately low stability prompt. **(b)** The FK-evolved 15-sequence prompt makes ProtGPT3-MSA more likely to generate ground-truth (Megascale) high stability variants, while making it less likely to generate low stability ones; paired t-test. **(c)** Against a compute-matched best-of-N baseline, FK performs equivalently for single mutants (99.1 vs. 98.9, $p = 0.17$), both quickly saturating the reachable ceiling. However, FK outperforms best-of-N for the larger space of double mutants; $p = 1.7\times10^{-4}$, , with the advantage being visible across the whole compute budget. **(d)** The same advantage appears for quadruple mutants from GB1 DMS (Wu et al., 2016), a different protein, assay and oracle, $p = 2.1\times10^{-5}$. **(e)** DPO alignment raises the fraction of generations with <25% low-complexity residues at every model scale, with the largest model achieving the largest benefit. **(f)** DPO alignment does so without collapsing the output distribution: sequence diversity is preserved and in fact slightly increased. Significance: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, ns = not significant.

7 Conclusion and Limitations

We introduce ProtGPT3, an open-source family of autoregressive pLMs spanning 112M to 10B parameters, together with ProtGPT3-MSA, a 112M-parameter model for homolog-conditioned generation. All models are fully integrated with the HuggingFace ecosystem and released with efficient training and generation code, lowering the barrier to customizable pLMs. A central finding is that homolog conditioned few-shot prompting can replace task-specific fine-tuning in target protein generation settings, providing an extra advantage in low data regimes. While still requiring homolog sequences, ProtGPT3-MSA reduces the burden of task-specific fine-tuning, as only a small set of related sequences is needed to generate family-consistent proteins without updating model weights. Experimentally, we show ProtGPT3 successfully generated novel active defluorinase enzymes in an extremely data scarce setting, increasing the number of active defluorinase enzymes from seven to eleven candidates. To augment the capabilities for controllability, we introduce a novel FK-SMC approach to steer MSA models toward a target property at inference time, demonstrating its effectiveness in increasing the stability and fitness of protein sequences. Finally, we show that large scale post-training alignment can reduce low-complexity generation while preserving sequence diversity, suggesting that, similarly to NLP models, pLMs benefit from large-scale alignment procedures to mitigate broad sampling biases beyond task-specific optimization. Despite these advances, our study has inherent limitations. While we tested defluorinase enzymes, both steering approaches (i.e., DPO and FK-SMC) remain not experimentally tested. Additionally, both approaches depend on the identification of reliable computational oracles for experimentally relevant protein properties, which remains an open challenge in the field. Despite these limitations, our results suggest that prompting, alignment, and inference-time compute represent a promising framework for controllable protein generation. By releasing weights, code, and dataset, we hope ProtGPT3 provides a useful resource for future work on scalable and controllable protein design.

AI USE STATEMENT

We used generative AI tools (Cursor, Anthropic Claude via Claude Code) as a coding assistant while building the benchmark code. They were not used for research ideation, experimental design or literature search. We used an LLM (ChatGPT) to correct the usage of English in some parts of the text. All numerical results come from running our code and not from a language model. All AI-assisted text was reviewed and edited by the authors, and we take responsibility for the final content of this work.

ETHICS STATEMENT

ProtGPT3 is built entirely from publicly available experimental sequences (UniRef90, GigaRef and OpenProteinSet Uniclust30); it involves no human subjects and no restricted-access data. All models, code, weights, and evaluation pipelines are published and publicly released, with attribution to the datasets and models they build on. We note the dual-use character of protein design research. The defluorinase case study targets the degradation of persistent environmental pollutants, and our models are general-purpose sequence generators with no pathogen-specific training data or capability. Nonetheless, ProtGPT3 has generative capabilities; for projects in any domain, active oversight during project initiation, experimental optimization, and deployment phases should be put in place to ensure safe usage and limitation of unintended harmful effects by the end users.

REPRODUCIBILITY STATEMENT

We publicly release all our models under a pseudonym for anonymity (https://huggingface.co/protgpt3) as well as training and inference code with the paper (see Appendix A.19). Our Appendix includes extensive details of all the experiments we ran, including all hyperparameter (Appendix A.17).

REFERENCES

Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Lukas Jarosch, Dan Berenberg, Ian Fisk, Andrew Watkins, Stephen Ra, Richard Bonneau, and Mohammed AlQuraishi. Openproteinset: Training data for structural biology at scale. *Advances in Neural Information Processing Systems*, 36: 4597–4609, 2023.

Shadab Ahmad, Leonardo Jose da Costa Gonzales, Emily H Bowler-Barnett, Daniel L Rice, Minjoon Kim, Supun Wijerathne, Aurélien Luciani, Swaathi Kandasaamy, Jie Luo, Xavier Watkins, et al. The uniprot website api: facilitating programmatic access to protein knowledge. *Nucleic acids research*, 53(W1):W547–W553, 2025.

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

Aadyot Bhatnagar, Sarthak Jain, Joel Beazer, Samuel C Curran, Alexander M Hoffnagle, Kyle S Ching, Michael Martyn, Stephen Nayfach, Jeffrey A Ruffolo, and Ali Madani. Scaling unlocks broader generation and deeper functional understanding of proteins. *bioRxiv*, pp. 2025–04, 2025.

Nathaniel Blalock, Srinath Seshadri, Agrim Babbar, Sarah A Fahlberg, Ameya Kulkarni, and Philip A Romero. Functional alignment of protein language models via reinforcement learning. *bioRxiv*, pp. 2025–05, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*, 26(5):689–691, 2010.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. xtrimopglm: unified 100-billion-parameter pretrained transformer for deciphering the language of proteins. *Nature Methods*, 22(5):1028–1039, 2025.

Xingyi Cheng, Bo Chen, Pan Li, Jing Gong, Jie Tang, and Le Song. Training compute-optimal protein language models. *Advances in Neural Information Processing Systems*, 37:69386–69418, 2024.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Ken A Dill, Kingshuk Ghosh, and Jeremy D Schmit. Physical limits of cells and proteomes. *Proceedings of the National Academy of Sciences*, 108(44):17876–17882, 2011.

Tuan Dinh, Seon-Kyeong Jang, Noah Zaitlen, and Vasilis Ntranos. Compressing the collective knowledge of esm into a single protein language model. *Nature Methods*, pp. 1–13, 2026.

Martin C Frith. A new repeat-masking method enables specific detection of homologous sequences. *Nucleic acids research*, 39(4):e23–e23, 2011.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. Liger-kernel: Efficient triton kernels for LLM training. 2025. URL https://openreview.net/forum?id=36SjAIT42G.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Sean R Johnson, Xiaozhi Fu, Sandra Viknander, Clara Goldin, Sarah Monaco, Aleksej Zelezniak, and Kevin K Yang. Computational scoring and experimental evaluation of enzymes generated by neural networks. *Nature biotechnology*, 43(3):396–405, 2025.

Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.

Achintya Kundu, Rhui Dih Lee, Laura Wynter, Raghu Kiran Ganti, and Mayank Mishra. Enhancing training efficiency using packing with flash attention. *arXiv preprint arXiv:2407.09105*, 2024.

Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, volume 487, pp. 545–574. Elsevier, 2011.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Lasse Middendorf and Noelia Ferruz. Generative ai for enzyme design and biocatalysis. *arXiv preprint arXiv:2602.03779*, 2026.

Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36: 64331–64379, 2023.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Silke I. Probst, Florian D. Felder, Victoria Poltorak, Ritesh Mewalal, Ian K. Blaby, and Serina L. Robinson. Enzymatic carbon–fluorine bond cleavage by human gut microbes. *Proceedings of the National Academy of Sciences*, 122(24):e2504122122, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International conference on machine learning*, pp. 8844–8856. PMLR, 2021.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. pp. 3505–3506, 2020.

Niklas Schmidinger, Lisa Schneckenreiter, Philipp Seidl, Johannes Schimunek, Pieter-Jan Hoedt, Johannes Brandstetter, Andreas Mayr, Sohvi Luukkonen, Sepp Hochreiter, and Günter Klambauer. Bio-xlstm: Generative modeling, representation and in-context learning of biological and chemical sequences. *arXiv preprint arXiv:2411.04165*, 2024.

Yaiza Serrano, Álvaro Ciudad, and Alexis Molina. Are protein language models compute optimal? *arXiv preprint arXiv:2406.07249*, 2024.

Damiano Sgarbossa, Cyril Malbranke, and Anne-Florence Bitbol. Protmamba: a homology-aware but alignment-free protein state space model. *Bioinformatics*, 41(6):btaf348, 2025.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Martin Steinegger, Markus Meier, Milot Mirdita, Harald Vöhringer, Stephan J Haunsberger, and Johannes Söding. Hh-suite3 for fast remote homology detection and deep protein annotation. *BMC bioinformatics*, 20(1):473, 2019.

Filippo Stocco, Maria Artigues-Lleixa, Andrea Hunklinger, Talal Widatalla, Marc Guell, and Noelia Ferruz. Guiding generative protein language models with reinforcement learning. *arXiv preprint arXiv:2412.12979*, 2024.

Filippo Stocco, Michele Garibbo, and Noelia Ferruz. Steering generative models for protein design: Aligning and conditioning strategies. *Current Opinion in Structural Biology*, 98:103250, 2026.

Timothy Truong Jr and Tristan Bepler. Poet: A generative model of protein families as sequences-of-sequences. *Advances in Neural Information Processing Systems*, 36:77379–77415, 2023.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444, 2023.

Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *BioRxiv*, pp. 2022–12, 2022.

Lawrence P Wackett. Confronting pfas persistence: Enzymes catalyzing c–f bond cleavage. *Trends in Biochemical Sciences*, 50(1):71–83, 2025.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Jude Wells, Alex Hawkins Hooker, Micha Livne, Weining Lin, David Miller, Christian Dallago, Nicola Bordin, Brooks Paige, Burkhard Rost, Christine Orengo, et al. Profam: Open-source protein family language modelling for fitness prediction and design. *bioRxiv*, pp. 2025–12, 2025.

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, pp. 2024–05, 2024.

Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *elife*, 5:e16965, 2016.

Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

Kevin K Yang, Sarah Alamdari, Alex J Lee, Kaeli Kaymak-Loveless, Samir Char, Garyk Brixi, Carles Domingo-Enrich, Chentong Wang, Suyue Lyu, Nicolo Fusi, et al. The dayhoff atlas: scaling sequence diversity for improved protein generation. *bioRxiv*, pp. 2025–07, 2025.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 MODEL ARCHITECTURE: SPARSE VS DENSE

Here we compare training and inference between a 112M-parameter Mixtral-style (sparse) mixture-of-experts (MoE) model (Jiang et al., 2024) and an equivalent 112M-parameter Llama3-style (dense) model (Grattafiori et al., 2024). Both models were trained on the same infrastructure (4× NVIDIA H100 GPUs) on 9.8B tokens, following the compute-optimal scaling law for AR pLMs from Cheng et al. (2024). Fig. 4a,b shows that the sparse Mixtral architecture achieves marginally worse validation perplexity (PPL; both forward and reverse) while training $\approx 1/3$ faster than the dense Llama3 architecture. Validation perplexity was computed on a held-out set of protein sequences, each sharing at most 50% sequence identity (0.8 coverage) with the training sequences. In addition, inference on the validation sequences is $\approx 1/2$ faster with the Mixtral architecture than with Llama3 on the same infrastructure (a single NVIDIA H100 GPU; Fig. 4c).

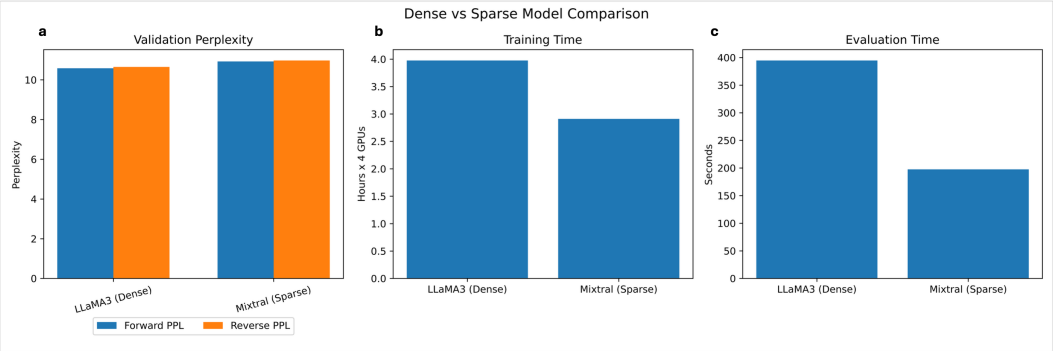

Figure 4: Model architecture: LLama3 (dense) vs Mixtral (sparse) model comparison in terms of a) validation perplexity on held-out protein sequences b) training time and c) inference

A.2 SINGLE-SEQUENCE MODELS PRETRAINING

All models were trained on publicly available datasets. Table 1 reports the number of protein sequences used to train each single-sequence model from the two datasets: UniRef90 (Ahmad et al., 2025) and GigaRef (Yang et al., 2025). The number of sequences per model follows compute-optimal scaling laws for sparse pLMs (Bhatnagar et al., 2025). The 112M and 1.3B models use 50% of their tokens from each dataset, and the 10B model uses 70% GigaRef and 30% UniRef90 tokens (Appendix A.3). During training, each sequence was randomly assigned a generation direction, either forward (N-to-C) or reverse (C-to-N), and the special token 1 or 2, respectively, was prepended to denote it. For maximal training efficiency, all models (including ProtGPT3-MSA) were trained with online mini-batch packing with Flash-Attention (Kundu et al., 2024).

Table 1: Number of protein sequences used to train each single-sequence model across the two datasets.

Model size	UniRef90	GigaRef	N. tokens
112M	15M	28M	9.8B
1.3B	64M	120M	43B
10B	174M	643M	170B

A.3 DATASET MIXING AND VALIDATION SETS

GigaRef includes some UniRef100-derived sequences, but they account for only 13M of its 243M clusters, so they are heavily underrepresented compared to metagenomic sequences. We tested two strategies for keeping UniRef sequences from being swamped by metagenomic data. In the first, each

batch contained an equal mix of tokens from GigaRef and UniRef90. In the second, we trained on GigaRef alone but applied a weighted loss that upweights the UniRef-derived sequences within it. We compared these against UniRef90-only and GigaRef-only baselines by training four 112M-parameter models. For evaluation, we built two validation sets, one from UniRef90 and one from GigaRef. We used MMseqs2 (Steinegger & Söding, 2017) to cap sequence identity between training and validation sequences at 50%, and we removed any cross-contamination between the UniRef90 and GigaRef splits. The 50/50 mixed-batch model gave the best trade-off across the two validation perplexities (Fig. 5), so we adopted this strategy for all other models. ProtGPT3-10B was the exception, trained on a 30/70 UniRef90/GigaRef mix because UniRef90 does not contain enough sequences to reach its compute-optimal number of tokens.

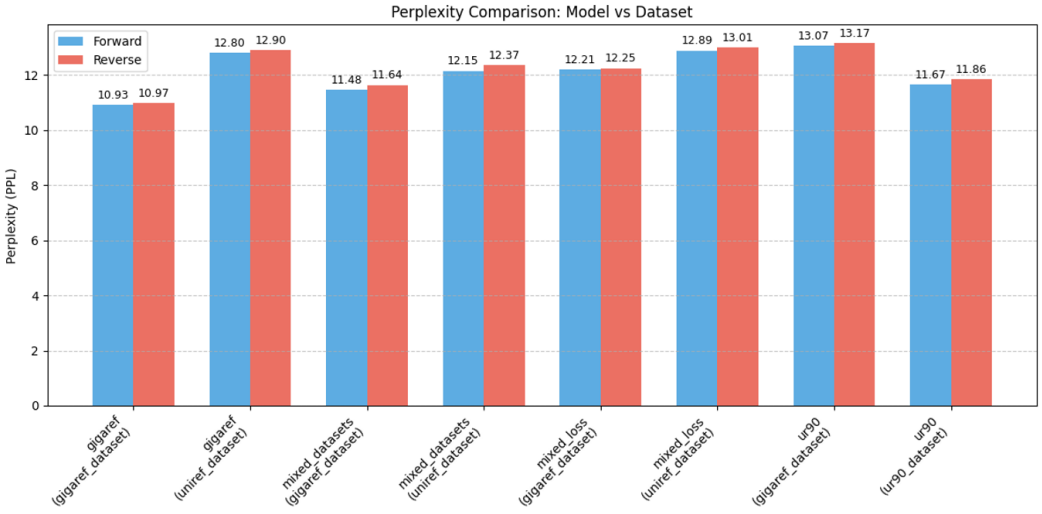

Figure 5: Validation perplexity on Gigaref and Uniref90 validation dataset across the four 112M-parameter models trained on: (1) GigaRef sequences only, (2) 50/50 mixed batches, (3) GigaRef with a weighted loss to up weight Uniref sequences, and (4) Uniref90 (ur90) sequences only.

A.4 ProtGPT3-MSA pretraining

ProtGPT3-MSA was initialized from the final ProtGPT3-112M checkpoint and trained on the OpenProteinSet Uniclust30 dataset (Ahdritz et al., 2023). We filtered out MSAs with fewer than 16 sequences and MSAs containing any sequence longer than 1024 amino acids, which guarantees that 16 sequences fit within the 16,384-token context. This reduced the 16M Uniclust30 MSAs to approximately 8.5M training MSAs. From each MSA, we sampled 16 sequences without replacement, concatenated them in random order, and separated them with a special separator token `<s>`. We used exactly 16 sequences even when the context could hold more, so that the number of sequences stayed balanced across training MSAs, preventing the model from learning spurious relationships between the number of sequences in the prompt and protein length. Each set of 16 sequences was randomly assigned to one of 4 modalities, defined by the generation direction (N-to-C or C-to-N) combined with the alignment modality. In the *aligned* modality, the sequences are passed to the model aligned, including gap tokens ("−"). In the *unaligned* modality, the full sequences are passed to the models, including insertions/deletions and without any gap. Two special tokens at the start of each input specify the format: `<gap>` for aligned and `<no_gap>` for unaligned. We repeated this sampling 15 times per MSA, reshuffling the sequence order each time, which gave approximately 560B training tokens in total.

A.5 Validation per-cluster perplexity analysis

To assess how uniformly model quality improves with scale, we evaluated each model perplexity at the level of 50% sequence clusters rather than computing a single aggregate perplexity across the entire validation set. We ran this analysis for held-out validation clusters, those sequences had

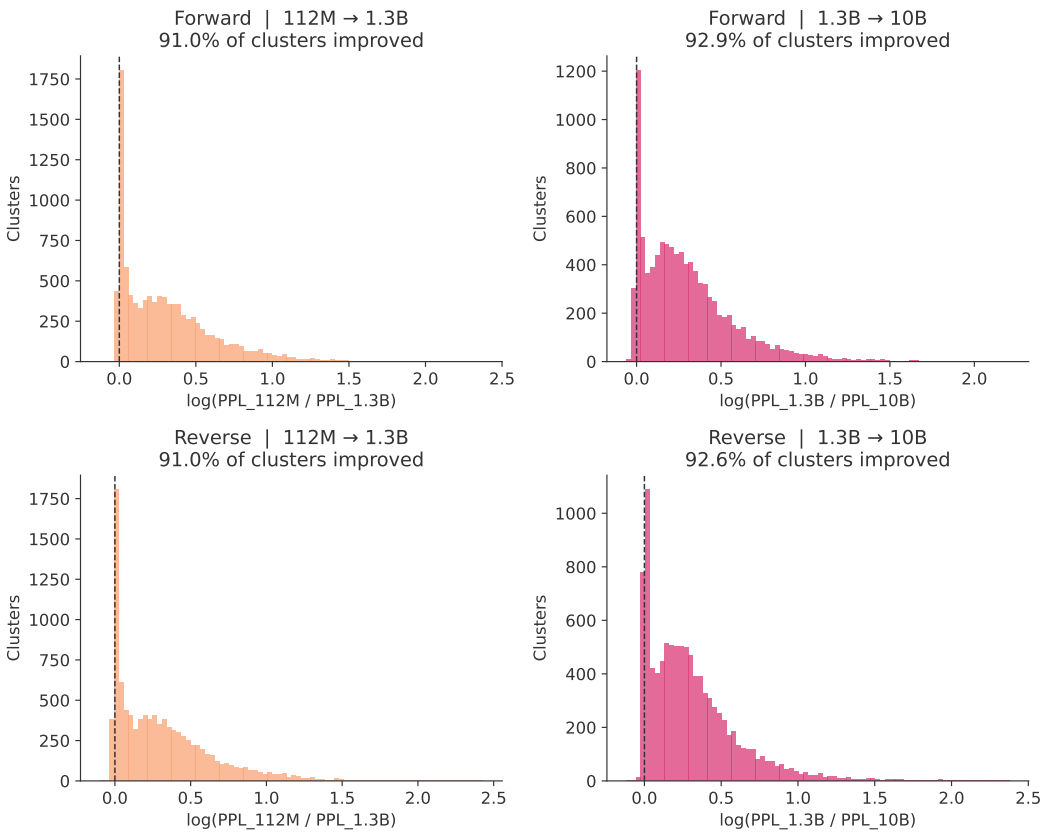

Figure 6: Per-cluster perplexity improvement across model scales. Log-ratio of per-cluster perplexity between adjacent model sizes, $\log(\frac{\text{PPL}_{\text{small}}}{\text{PPL}_{\text{large}}})$, for forward (N-to-C) (top) and reverse (C-to-N) terminal (bottom) direction. Values greater than zero indicate clusters where the larger model assigns lower perplexity. Both scaling steps improve perplexity on over 91% of clusters, with the 1.3B→10B transition showing a broader distribution of gains, reflecting more uniform improvement across the sequence space.

at most 50% sequence identity to the training set of each model. Concretely, for each cluster in the held-out validation set we computed the mean token-level negative log-likelihood across all member sequences, evaluated for both forward (N-to-C) and reverse (C-to-N) terminal direction, and converted this to a per-cluster perplexity. This cluster-level view allows us to distinguish between a model that improves uniformly across the sequence space and one that improves only on clusters it already models well, a distinction that is invisible when reporting a single validation average.

Fig. 6 shows the log-ratio of per-cluster perplexity (PPL) between adjacent model sizes, $\log(\frac{\text{PPL}_{\text{small}}}{\text{PPL}_{\text{large}}})$ for both forward (N-to-C) and reverse (C-to-N) terminal direction. Values to the right of zero indicate clusters where the larger model assigns lower perplexity, i.e. models the sequences better. Both transitions, 112M to 1.3B and 1.3B to 10B, show strongly right-skewed distributions, with over 91% of clusters improving at each step and in both orientations. Notably, the 1.3B→10B histograms display a broader, more uniformly spread distribution compared to the sharper spike near zero seen for 112M→1.3B, indicating that the gain from the largest model is more evenly distributed across clusters rather than concentrated on a small subset. The near-identical results for forward and reverse orientations confirm that the bidirectional training objective is learning consistent representations regardless of sequence direction.

A.6 PROTEINGYM BENCHMARKING

In Table 2, we report the results of benchmarking each model on ProteinGym (Notin et al., 2023). The mean Spearman correlation was computed using the official ProteinGym script, `performance_DMS_benchmarks.py`. The median, standard deviation, minimum, and maximum were computed over the DMS-level Spearman correlations. Assemblies were constructed from sets of 16 orthologous sequences randomly sampled from the precomputed MSAs provided by ProteinGym. Each sequence was evaluated in both the forward and reverse directions using the corresponding direction tokens. The resulting log-likelihoods were averaged across directions. The change of fitness determined by the mutation (Δf) was computed as a paired log-likelihood difference between each mutant sequence and its corresponding wild-type sequence. For each replicate, the mutant and wild-type sequences were each inserted as the 16th sequence and scored using the same set of 15 orthologous context sequences, ensuring a paired comparison. For each DMS assay, we evaluated a random subsample of up to 1,000 variants. The final fitness score was obtained by averaging the paired differences over 10 replicates.

For each variant, we used a paired scoring procedure in which the mutant and wild-type sequences were evaluated using the same sampled assembly context. Let $d \in \{\mathrm{fwd}, \mathrm{rev}\}$ denote the sequence direction. We first computed the assembly-averaged sequence score

$$s(x) = \frac{1}{2} \sum_{d \in \{\mathrm{fwd},\mathrm{rev}\}} \frac{1}{L} \sum_{i=1}^{L} \log p_\theta(x_i \mid x_{<i}, d)\,.$$

In case of ProtGPT3-MSA, we have also averaged over 10 different samples of orthologous sequences. The variant fitness score was then defined as the paired difference between the mutant and wild-type scores:

$$\Delta f = \log \frac{p_\theta(x^{\mathrm{mut}})}{p_\theta(x^{\mathrm{WT}})}.$$

Table 2: Spearman correlation (ρ) between model log-likelihoods and experimental fitness scores on the ProteinGym DMS benchmark.

Model	DMS Type	N	Mean ρ	Median ρ	Std	Min	Max
ProtGPT3-10B	All	283	0.407	0.435	0.186	-0.122	0.795
	Substitution	217	0.357	0.390	0.179	-0.143	0.773
	Indels	66	0.457	0.481	0.194	-0.102	0.822
ProtGPT3-1.3B	All	283	0.386	0.403	0.204	-0.173	0.762
	Substitution	217	0.322	0.360	0.191	-0.159	0.745
	Indels	66	0.450	0.446	0.217	-0.187	0.780
ProtGPT3-112M	All	283	0.271	0.279	0.201	-0.263	0.713
	Substitution	217	0.235	0.225	0.198	-0.379	0.739
	Indels	66	0.306	0.333	0.204	-0.147	0.687
ProtGPT3-MSA	All	283	0.391	0.426	0.192	-0.046	0.800
	Substitution	217	0.357	0.369	0.163	-0.057	0.805
	Indels	66	0.425	0.483	0.222	-0.036	0.796
ProtGPT3-MSA(ensemble)	All	282	**0.422**	0.478	0.300	0.083	0.817
	Substitution	216	**0.384**	0.396	0.162	-0.065	0.808
	Indels	66	**0.460**	0.565	0.205	-0.102	0.827

A.7 PROTGPT3-MSA: COMPARISON TO OTHER MODELS

To position ProtGPT3-MSA relative to existing family-conditioned generation approaches, we evaluated it in two complementary settings. First, we compared six homolog-conditioned models on 25 randomly selected family MSAs, providing each model with the same 15 unaligned homologs and generation budget. This benchmark evaluates novelty relative to the prompt, agreement with the held-out family distribution, structural confidence, and computational efficiency. ProtGPT3-MSA

achieves the fastest generation and the highest mean pLDDT while remaining competitive with the strongest baselines in HMMER bitscore and column-wise KL divergence. It also maintains low exact-copy and one-mutant rates (Table 3).

We additionally compared ProtGPT3-MSA with ZymCTRL and ProGen-1, which use discrete control codes rather than homolog prompts. Because the two models use different conditioning vocabularies, they cannot be evaluated on a single shared family set. ZymCTRL was evaluated on 15 EC families, whereas ProGen-1 was evaluated on a separate set of 15 EC-pinned homologous UniProt families. Each comparison used 20 generations per model and family. Compared with ZymCTRL, ProtGPT3-MSA increased mean pLDDT from 58.5 to 74.9 and CLEAN exact-match rate from 7.3% to 26.0%. Compared with ProGen-1, it increased mean pLDDT from 36.8 to 72.6 and CLEAN exact-match rate from 8.0% to 84.0%. ProtGPT3-MSA also required substantially less peak GPU memory than either baseline. Its generation speed was slightly lower than that of ZymCTRL (2.05 versus 1.71 seconds per sequence), but substantially higher than that of ProGen-1 (1.84 versus 11.68 seconds per sequence). Results should therefore be compared within each matched benchmark rather than across the two evaluation sets.

Table 3: Comparison of MSA-conditioned generators prompted with the same 15 unaligned homologs on 25 family MSAs (32 generations each). Exact-copy / 1-mutant rates use ungapped Hamming distance to the nearest prompt. HMMER bitscores and column KL use an HMM / match-state profile fit on held-out MSA sequences (KL: median across MSAs, mean in parentheses). Mean pLDDT is from ESMFold on sequences truncated to 400 aa. Inference time is mean wall-clock seconds per generated sequence on one NVIDIA GH200 (generation only); peak memory is mean peak GPU memory during generation.

Method	Exact copy ↓	1-mutant ↓	Mean d	HMMER bits ↑	Column KL ↓	Mean pLDDT ↑	s/seq ↓	Peak mem. ↓
ProtGPT3-MSA	0.38%	0.75%	82.8	98.6	0.286 (0.649)	62.2	0.80	313 MB
PoET	2.25%	7.62%	86.7	102.6	0.277 (0.327)	61.6	0.89	588 MB
ProtMamba	0.00%	0.00%	98.1	55.9	0.360 (0.386)	56.7	1.13	295 MB
Bio-xLSTM	0.25%	0.25%	86.7	87.4	0.254 (0.274)	60.4	1.97	1.37 GB
ProFam	0.50%	0.25%	91.4	76.7	0.256 (0.278)	58.6	2.14	685 MB
Dayhoff-MSA	0.50%	0.00%	91.8	64.3	0.281 (0.295)	56.3	11.06	6.48 GB
Held-out real	—	—	—	95.3	0.204 (0.222)	—	—	—

Table 4: Comparison with ZymCTRL on 15 EC families, using 20 generations per model and EC. ProtGPT3-MSA is prompted with up to 15 unaligned homologs from the target EC, whereas ZymCTRL is zero-shot conditioned on the EC string. Mean pLDDT is calculated with ESMFold on sequences truncated to 400 aa. CLEAN columns report exact four-level EC agreement, first-three-level agreement, and top-level class agreement. Inference time is the mean wall-clock time per generated sequence after a one-sequence warmup, using eight timed sequences per label. Generation speed and peak GPU memory were measured on one NVIDIA GH200.

Method	Mean pLDDT ↑	pLDDT ≥ 70 ↑	pLDDT ≥ 80 ↑	CLEAN exact ↑	CLEAN 3-level ↑	CLEAN 1-level ↑	s/seq ↓	Peak mem. ↓
ProtGPT3-MSA	74.9	80.3%	41.0%	26.0%	71.0%	86.7%	2.05	815 MB
ZymCTRL	58.5	36.7%	16.7%	7.3%	50.3%	66.3%	1.71	5.15 GB

Table 5: Comparison with ProGen-1 on 15 EC-pinned homologous UniProt families, using 20 generations per model and family. ProtGPT3-MSA is prompted with up to 15 unaligned homologs, whereas ProGen-1 is conditioned on a UniProt keyword set selected for the family. Mean pLDDT is calculated with ESMFold on sequences truncated to 400 aa. CLEAN agreement is evaluated against the family's unique four-level EC. Inference time is the mean wall-clock time per generated sequence after a one-sequence warmup, using eight timed sequences per label. Generation speed and peak GPU memory were measured on one NVIDIA GH200.

Method	Mean pLDDT ↑	pLDDT ≥ 70 ↑	pLDDT ≥ 80 ↑	CLEAN exact ↑	CLEAN 3-level ↑	CLEAN 1-level ↑	s/seq ↓	Peak mem. ↓
ProtGPT3-MSA	72.6	66.7%	48.0%	84.0%	88.3%	91.7%	1.84	878 MB
ProGen-1	36.8	3.7%	2.7%	8.0%	9.0%	23.0%	11.68	4.96 GB

A.8 MEGASCALE STABILITY ORACLE TRAINING

To provide the stability reward signal used during stability-aware protein generation and FK steering, we train a separate oracle model to rank a wild-type protein's variants by their predicted change in stability upon mutation ($\Delta\Delta G$). We fine-tune ProtGPT3-112M, using Group Relative Policy Optimization (GRPO), with $\Delta\Delta G_{\mathrm{ML}}$ from the most curated split (dataset 3) of MegaScale protein stability dataset (Tsuboyama et al., 2023) as the scalar reward, grouped by wild-type identity. The oracle is trained to increase its own sequence log-likelihood for more stable variants and decrease it for less stable ones, so that at its log-likelihood serves as a stability proxy without requiring a separate regression head. Training uses a KL coefficient of $\beta = 0.1$, a learning rate of 1×10^{-4} with a cosine schedule and 5% warmup, and a batch size of 50 sequences. The validation split was created by randomly sampling 10% of all MegaScale WTs with all their associated variants, ensuring no overlap between train/val WTs.

Training was configured for up to 2 epochs, but was stopped early, after approximately 1.16 epochs, once the Spearman correlation between the oracle's log-likelihood and ground-truth $\Delta\Delta G_{\mathrm{ML}}$ on the held-out validation split had plateaued (i.e., $\rho = 0.73$ on the training split and $\rho = 0.35$ on the validation split) (see Figure 7). Note, we use the oracle's own training split for FK steering because the experiment's goal is to test whether the generator can be steered by a given reward signal, not to re-test the oracle's generalization (that's already assessed separately via the train/val Spearman gap). Since steering quality is bottlenecked by how reliable the reward signal is on the sequences being searched, we need the oracle to be maximally accurate there — which is guaranteed on training data ($\rho = 0.73$) rather than validation data ($\rho = 0.35$), where using the noisier val-generalization signal would confound "does steering work" with "does the oracle generalize."

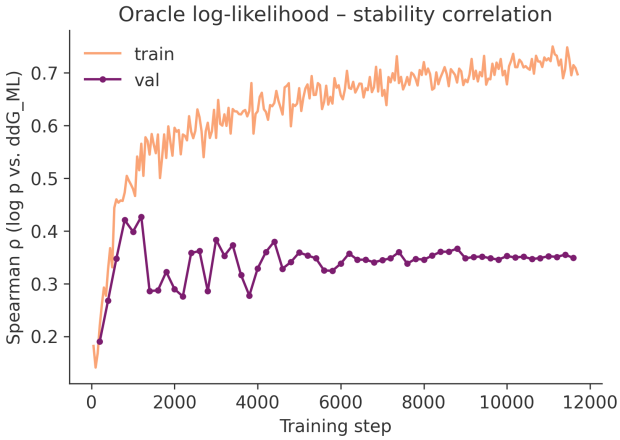

Figure 7: Spearman correlation between the oracle's sequence log-likelihood and ground-truth $\Delta\Delta G_{\mathrm{ML}}$, on the training (orange) and validation (purple) splits, across GRPO training steps.

A.9 ProtGPT3-MSA stability-prompting

We test whether few-shot prompting can steer ProtGPT3-MSA toward protein variants with a target stability profile. We first train an oracle on the most curated split (dataset 3) of MegaScale protein stability dataset (Tsuboyama et al., 2023) (see Appendix A.8). Next, we randomly sampled ten DMS libraries from the oracle's training set (Tsuboyama et al., 2023) and, for each, constructed 4 prompt sets corresponding to different (oracle-predicted) stability percentile groups (0-25%, 25-50%, 50-75%, 75-100%; Q1-Q4), with each set containing 15 variants from the same percentile. Each prompt set was then used to generate 3,000 sequences with ProtGPT3-MSA (across temperatures $= \{0.8, 0.9, 1.0\}$), which were then scored with the oracle to assign each a stability percentile. As reported in the main text, we find that when ProtGPT3-MSA is prompted with sequences from a higher stability percentiles, it generates sequences higher in stability (spearman correlation of 0.85 and $p < 0.001$). We further validate this effect on ground-truth data, by showing ProtGPT3-MSA assigns higher likelihood to held-out ground-truth sequences matching the prompted percentile than to sequences from other percentiles, an effect that disappears under a scrambled-label control.

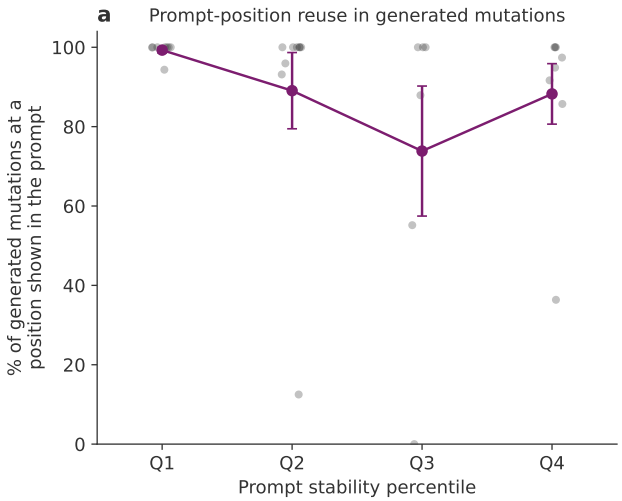

Figure 8: Generated mutations in ProtGPT3-MSA stability-prompting largely reuse positions shown in the prompt. Percentage of generated mutation events landing on a sequence position already mutated in the 15-example prompt, by prompt stability percentile. Points: per-WT values; line and markers: mean ± s.e.m. across WTs.

To investigate this effect, we asked whether a generated sequence's mutations tend to land on the same positions already mutated across its 15 prompt examples, rather than on new, percentile-specific positions elsewhere in the protein. For each WT and prompt percentile, we built a majority-vote consensus sequence from the same pool of MegaScale variants used to define the oracle stability percentiles, identified every position mutated relative to this consensus across the 15 prompt sequences ("prompt-shown" positions), and classified every mutated position in the corresponding generated sequences (one "event" per mutated position) as landing on a prompt-shown position or not. Fig 8 reports the resulting percentage of generated mutation events landing on a prompt-shown position, per prompt percentile, averaged (unweighted mean, one value per WT) across WTs. This percentage is high across all four percentiles (mean ± s.e.m. across WTs: Q1 $99.3 \pm 0.7\%$; Q2 $89.1 \pm 9.6\%$; Q3 $73.8 \pm 16.4\%$; Q4 $88.3 \pm 7.6\%$), showing that the model's stability-conditioned generations largely reuse mutation sites already demonstrated by the prompt rather than discovering new, percentile-specific sites. This does not undercut the steering result above: the model still has to choose which residue to substitute at each reused position, and, as shown above, its generated sequences are typically more stable than the prompt sequences themselves for Q1–Q3, rather than simple copies of them. Instead, it suggests that few-shot prompting steers generation mainly by indicating where stability-relevant substitutions tend to occur for a given WT, with which substitution to make then guided by whatever the model already knows about stability at that position.

A.10 Evaluation pipeline SFT versus few-shot prompting on held-out validation MSAs

We build an evaluation pipeline to compare how few-shot prompting of ProtGPT3-MSA compares to SFT of single-sequence models (ProtGPT3-112M and ProtGPT3-10B) for conditional protein generation on 25 target protein family randomly sampled from ProtGPT3-MSA held-out validation set (see Appendix A.11 for details on the sampling procedure). For SFT, each target family is treated as an independent fine-tuning task: we construct per-family train and validation splits, fine-tune the model separately on each, and select the model checkpoint that achieves the lowest validation perplexity for sequence generation after identifying the optimal learning rate for each model (see Appendix A.10.1). For few-shot prompting of ProtGPT3-MSA, we randomly sample 15 sequences from each target family to prompt the model at generation time without any parameter updates. When comparing few-shot prompting of ProtGPT3-MSA to SFT, all ProtGPT3-MSA sequences were generated in *unaligned* modality to ensure a fair comparison (i.e., single-sequence models were not trained to process aligned sequences). However, we include a comparison between *unaligned* and *aligned* generation in Appendix Fig. 11 for a subset of 10 held-out validation MSAs. For both SFT and few-shot prompting, generated sequences are evaluated against the reference family across four complementary metrics (i) maximum sequence identity, (ii) maximum structure similarity, (iii) HMM profile comparison and positional (iv) KL-divergence comparison, each described in detail in the following subsections.

A.10.1 SFT training protocol

SFT on held-out validation MSAs We use a tight, two-stage protocol designed to give each SFT model its best shot:

(i) A per-model-size learning-rate sweep (LR $\in \{1\times10^{-6}, 1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 7\times10^{-5}\}$) on one random held-out family, fixed thereafter;

(ii) Per-family fine-tuning for up to 50 epochs (112M) / 25 epochs (10B), with the lowest-validation-loss checkpoint used for generation (early stopping).

Table 6 shows that the train loss continues to improve well past the epoch where validation loss peaks for both models, and even the last family to peak its validation loss (max best eval epoch: 25/14) does so with over half the training budget unused. This confirms neither SFT model was undertrained; more epochs would only add overfitting. Additionally, because we select the checkpoint at the epoch of lowest validation loss rather than the final epoch, each model is taken at the point of best generalization, before validation loss begins to rise, ensuring the reported models did not overfit.

Table 6: SFT training and evaluation results by model size.

Model	Optimal LR	Mean Best Train Loss (epoch)	Mean Best Eval Loss (epoch)	Max Best Eval Epoch
SFT-112M	5e-5	0.9973 (43.29)	1.9310 (15.32)	25 out of 50
SFT-10B	2e-5	0.1484 (23.62)	1.7683 (6.52)	14 out of 25

A.10.2 Maximum Sequence Identity

To assess the similarity of generated sequences to the reference protein family, we computed the maximum fractional sequence identity between each generated sequence and all reference sequences using MMseqs2 `easy-search` (Steinegger & Söding, 2017). For each generated sequence, we retained the highest identity match across all reference targets, assigning zero identity to sequences that produced no alignment, so that summary statistics reflect the full generated set. We report the mean maximum sequence identity across all generated sequences within each condition, where lower values indicate greater dissimilarity from known family members.

A.10.3 Maximum Structure Similarity

To assess the structural similarity of generated sequences to the reference protein family, we first folded all generated sequences using ESMFold (Lin et al., 2023) and then computed the maximum

TM-score between each predicted structure and all reference structures within the same protein family using Foldseek `easy-search` (Van Kempen et al., 2024). For each generated structure, we retained the highest TM-score across all reference targets, normalised by target length, and report the mean maximum TM-score across all generated sequences within each condition. For each predicted structure, we also report ESMFold pLDDT estimates.

A.10.4 HMM PROFILE COMPARISON

To assess how well generated sequences capture the statistical properties of a target protein family, we compare Hidden Markov Model (HMM) profiles built from generated sequences against profiles built from reference family alignments. The pipeline consists of three steps.

First, the generated sequences must be placed into the coordinate frame of the reference family MSA before any column-wise comparison is meaningful. We use MAFFT (Katoh & Standley, 2013) to insert generated sequences into the reference family MSA, while preserving the original alignment length and prevent the reference columns from being perturbed. This ensures that position i in the resulting alignment corresponds to the same residue position across both reference and generated sequences.

Second, we build two HMM profiles using the HH-suite (Steinegger et al., 2019). The reference profile is constructed directly from the reference MSA. The generated profile is constructed from the aligned generated sequences extracted in Step 1. Each profile encodes per-column amino acid emission probabilities as a position-specific frequency matrix (PSFM), capturing the residue composition and evolutionary variation at each position in the alignment. A minimum coverage threshold is applied: columns where fewer than 30% of reference sequences contribute a non-gap residue are excluded, as such columns provide unreliable frequency estimates.

Third, the two profiles are compared using again the HH-suite (Steinegger et al., 2019). This produces several complementary similarity metrics: the alignment probability (Prob), E-value, and bit score. Collectively, these reflect how structurally similar the two profiles are (i.e., the degree to which the generated sequences reproduce the column-wise amino acid distributions of the reference family). A high bit score, combined with a low E-value, indicates that the generated sequences faithfully capture the positional residue preferences of the target family.

A.10.5 POSITIONAL KL-DIVERGENCE COMPARISON

To quantify how closely the residue distributions of generated sequences match those of a reference protein family at each alignment position, we compute the column-wise Kullback–Leibler (KL) divergence between position-specific frequency matrices (PSFMs) derived from generated and reference sequences respectively. The pipeline consists of three steps.

First, as in the HMM profile comparison, generated sequences are first inserted into the reference MSA coordinate frame using MAFFT (Katoh & Standley, 2013), ensuring that column indices are shared between the reference and generated alignments.

Second,a PSFM of shape $(L \times 20)$ is computed independently for the reference MSA and the aligned generated sequences, where L is the alignment length and 20 corresponds to the standard amino acid alphabet (i.e., gaps are excluded from the KL computation). For each column j, residues are counted across all sequences, gap characters are ignored, and a small uniform pseudocount $\epsilon = 10^{-6}$ is added to every amino acid bin before normalising to a probability distribution. This prevents undefined logarithm terms at unobserved amino acids without materially distorting well-sampled columns. Columns where fewer than 30% of reference sequences contribute a non-gap residue are excluded from the analysis, as such positions carry unreliable frequency estimates.

Third, for each retained column j, we compute the KL divergence from the generated distribution P_j to the reference distribution Q_j:

$$D_{\mathrm{KL}}(P_j \| Q_j) = \sum_{a=1}^{20} P_j(a) \log \frac{P_j(a)}{Q_j(a)}, \tag{1}$$

where the sum runs over the 20 standard amino acids. The mean KL divergence

$$\bar{D} = \frac{1}{|J|} \sum_{j \in J} D_{\mathrm{KL}}(P_j \| Q_j) \tag{2}$$

over all retained columns J provides a single scalar summary of how faithfully the generated sequences reproduce the positional residue preferences of the reference family. Lower values indicate closer agreement with the reference statistics, with $\bar{D} = 0$ only when the generated and reference distributions are identical at every retained position.

A.10.6 Statistical methods

Here we details the statistical tests used to assess the performance of few-shot prompting and SFT across data regimes and model sclale (i.e., Fig. 2 top-row in the main text).

Depth-scaling correlations and confidence intervals (Fig. 2a) For each model and metric we compute Spearman's ρ between MSA depth (n sequences in the training MSA) and the metric, across the 25 held-out protein-family clusters, with a 95% percentile bootstrap CI (10,000 resamples with replacement of the 25 paired (depth, metric) observations). fident and TM-score are both maximum-of-N statistics – a deeper training MSA yields more generated candidate sequences, which mechanically inflates a max-of-N score independent of true generation quality – so both are first divided by that cluster's own train-to-validation ceiling (the mean best score obtained when the family's *real* training sequences are compared against its held-out validation sequences) before computing ρ. HH-score is not a max-of-N statistic and is used raw.

We exclude KL-divergence from this depth-correlation panel: this style of per-column KL metric is known to scale with the number of sequences in the reference MSA (see Appendix A.10.5), as is commonly the case in the literature (e.g., Wells et al., 2025), so we do not treat its depth-scaling as a reliable model-quality signal; it is still reported in the win-rate, interaction, and control panels/tables.

Results. fident correlates strongly with depth for both SFT models ($\rho = 0.78$ [95% CI 0.54, 0.89] for SFT-112M; $\rho = 0.74$ [0.52, 0.86] for SFT-10B), and HH-score follows the same trend, clearly for SFT-112M ($\rho = 0.49$ [0.06, 0.79]) and more weakly for SFT-10B ($\rho = 0.39$ [-0.03, 0.70]); TM-score shows no depth effect for either model ($\rho = 0.15$ [-0.26, 0.54] and $\rho = 0.11$ [-0.26, 0.47], respectively). ProtGPT3-MSA shows no depth correlation on any metric (fident $\rho = -0.06$ [-0.47, 0.34]; TM-score $\rho = -0.29$ [-0.67, 0.15]; HH-score $\rho = 0.14$ [-0.29, 0.51]), consistent with its fixed 15-sequence retrieval context being unaffected by the underlying training-MSA depth.

Win-rate significance (Fig. 2b) For each of the 25 test MSAs we identify, per metric, whichever of the three models achieves the best score. We run a per-metric two-sided exact binomial sign test of the pre-identified best model's win count against chance $= 1/2$, i.e. whether the winning model wins strictly more than half of the 25 comparisons against "the other two combined." This is the test starred in Fig. 2b.

Results. ProtGPT3-MSA is the winning model for every metric, with a win rate significantly above chance for fident (18/25, $p = 0.043$), TM-score (18/25, $p = 0.043$), and HH-score (19/25, $p = 0.015$), but not for KL-divergence (14/25, $p = 0.69$).

Depth-regime interaction test (Fig. 2c) For a given metric and model pair (e.g. MSA vs. SFT-10B), we compute the paired per-cluster difference across the 25 families. We then split them by a fixed threshold of 340 Neff sequences into a low-depth group ($n = 12$) and a high-depth group ($n = 13$), and bootstrap each group's mean difference separately (10,000 resamples). The interaction effect is $\Delta_{\mathrm{interaction}} = \Delta_{\mathrm{low}} - \Delta_{\mathrm{high}}$, encoding how much larger the model-vs-model advantage is in the low-depth regime than in the high-depth regime. The significance test comes from resampling the two regimes independently and taking the empirical two-sided p-value of the bootstrap difference distribution. Note, we use this nonparametric bootstrap rather than a classical two-way ANOVA F-test because the data violate that test's assumptions: only 25 families split into small, unbalanced, data-driven groups (12 vs. 13); paired rather than independent observations (the same 25 families are scored under every model); and metric-specific distributional issues (KL is right-skewed with mean-scaling variance; fident/TM-score are bounded proportions; HH-score is heavy-tailed). This also keeps the methodology consistent with the bootstrap CIs used everywhere else in the analysis.

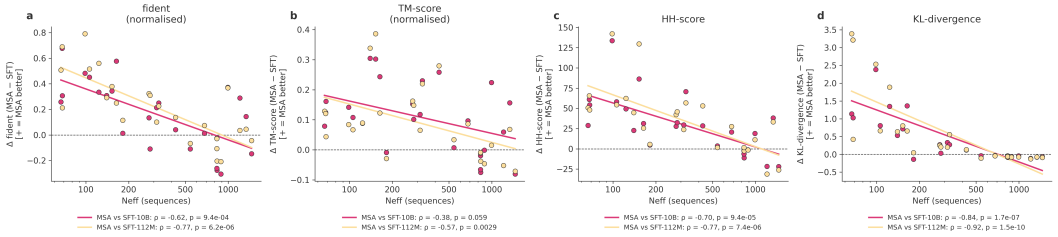

Figure 9: **The MSA model's advantage over SFT shrinks with MSA depth.** Each point is one of the 25 held-out test MSAs. The y-axis shows the per-MSA gap between the MSA model and SFT-10B (pink) or SFT-112M (yellow), signed so that positive means the MSA model is better; the dashed line marks no difference. The reported statistic is Spearman's ρ, which is rank-based and therefore unaffected by the log x-axis. **a**, fident (normalised): the gap in best-hit sequence identity decreases with N_{eff} ($\rho = -0.62$, $p = 9.4 \times 10^{-4}$ vs. SFT-10B; $\rho = -0.77$, $p = 6.2 \times 10^{-6}$ vs. SFT-112M). **b**, TM-score (normalised): same trend for best-hit structural similarity, but weaker ($\rho = -0.57$, $p = 0.0029$ vs. SFT-112M; not significant vs. SFT-10B, $\rho = -0.38$, $p = 0.059$). **c**, HH-score: the profile-level (HMM–HMM) similarity gap decreases with N_{eff} ($\rho = -0.70$, $p = 9.4 \times 10^{-5}$ vs. SFT-10B; $\rho = -0.77$, $p = 7.4 \times 10^{-6}$ vs. SFT-112M). **d**, KL-divergence (gap sign-flipped, since lower is better): the strongest trend ($\rho = -0.84$, $p = 1.7 \times 10^{-7}$ vs. SFT-10B; $\rho = -0.92$, $p = 1.5 \times 10^{-10}$ vs. SFT-112M), with the MSA model's advantage largest for shallow MSAs and reaching parity at $N_{\text{eff}} \approx 500$–$1000$. *Normalisation* (**a**, **b**): fident and TM-score are best-hit (max-over-sequences) statistics, so they grow with the number of sequences searched regardless of generation quality. To remove this depth effect, each MSA's mean best-hit fident and TM-score is divided by the mean best-hit value that the real training sequences of the same MSA achieve against its held-out validation sequences. HH-score and KL-divergence are not max-over-sequences statistics and are used raw. The number of effective sequences (Neff) for each MSA (x-axis) is computed as described by (Morcos et al., 2011) with re-weighting based on 80% sequence identity clustering.

Results. We found the interaction effect to be significant across all 4 metrics, see Table 7. Additionally, Fig 9 show the correlation between each performance gap relative to the two SFT models and Neff of each MSA, highlighting MSA model's advantage over SFT shrinks with MSA depth.

Null controls (Fig. 2d). *Bootstrap-MSA*: for each cluster, we subsample 20% of the pooled real (train + validation) unaligned reference sequences, align them with MAFFT exactly as for generated sequences, and score them against the full reference MSA with the same HH-score/KL pipeline. No generative model is involved; this measures the best achievable score given the family's own natural sequence diversity and alignment procedure, i.e. an upper bound (or, for KL, a lower/best-achievable bound). *Scrambled-MSA*: each cluster's generated sequences are instead scored against the reference MSA of a different, cyclically-shifted cluster ($\text{cluster}_i \to \text{cluster}_{i+1 \bmod 25}$). If generated sequences carry real family-specific signal, they should score far worse against this mismatched reference than against their own family's reference, giving a lower bound (or, for KL, an upper/worst-case bound).

Results. All three models separate clearly from the scrambled control (mean HH-score 24.0 ± 4.6; mean KL 3.95 ± 0.42) and land much closer to the bootstrap ceiling (mean HH-score 194.2 ± 16.8; mean KL 0.106 ± 0.015): SFT-112M (140.0 ± 14.1 HH-score, 1.05 ± 0.24 KL), SFT-10B (145.3 ± 13.9 HH-score, 0.96 ± 0.20 KL), and ProtGPT3-MSA (174.6 ± 15.1 HH-score, 0.45 ± 0.06 KL) – with ProtGPT3-MSA the closest of the three to the bootstrap ceiling on both metrics.

A.11 Sampling of held-out validation MSAs for conditional protein generation comparison

To ensure the validation set spans a broad range of MSA depths, we employed a stratified log-spaced sampling procedure. All eligible MSAs were first filtered to a depth range of 100 to 2000 sequences, and the resulting pool was partitioned into five logarithmically-spaced bins spanning that range (see Table 8). A fixed number of MSAs were then drawn uniformly at random from each bin, with any shortfall in underpopulated bins redistributed to the remaining ones. This procedure guarantees that the sampled validation MSAs are evenly distributed across depth magnitudes, from shallow MSAs

Table 7: Depth-regime interaction test for the MSA-vs-SFT-10B comparison, per metric (Fig. 2c; see Appendix A.10.6 for the test). Δ is defined as MSA $-$ SFT-10B; for KL (lower is better) a negative Δ means MSA is better, unlike the other three metrics where positive $=$ MSA better. Low-depth: $n = 12$ clusters < 500 sequences; high-depth: $n = 13$ clusters ≥ 500 sequences.

Metric	Low-depth Δ [95% CI]	High-depth Δ [95% CI]	Interaction Δ [95% CI], p
fident	0.315 [0.195, 0.435]	0.015 [−0.102, 0.131]	0.300 [0.127, 0.467], $p < 0.001$
TM-score	0.150 [0.102, 0.202]	0.070 [0.008, 0.133]	0.080 [0.000, 0.161], $p = 0.050$
HH-score	49.27 [32.40, 69.06]	10.93 [−2.34, 25.40]	38.34 [15.88, 62.34], $p < 0.001$
KL	−1.058 [−1.621, −0.577]	−0.007 [−0.090, 0.060]	−1.051 [−1.610, −0.554], $p < 0.0001$

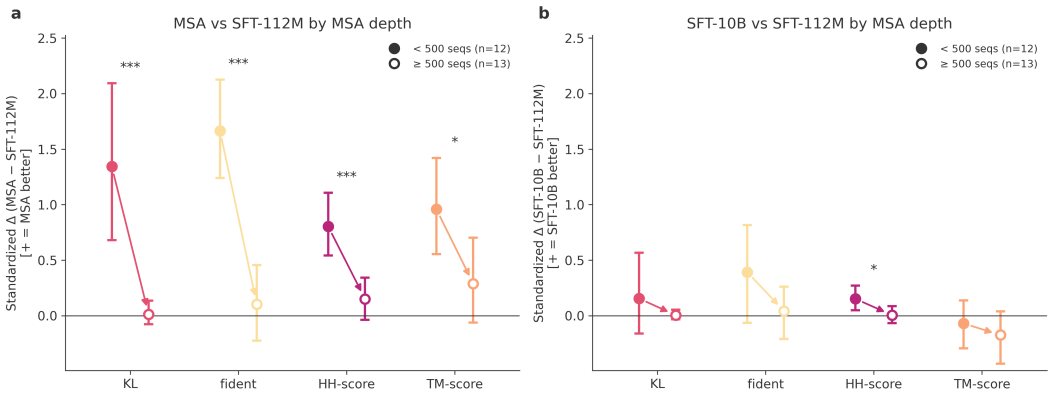

Figure 10: **Depth-regime comparison for the remaining model pairs.** Mean standardized score difference with 95% bootstrap CIs for **(a)** ProtGPT3-MSA vs. SFT-112M and **(b)** SFT-10B vs. SFT-112M (positive $=$ first model better), separately for low-depth (approximately < 500 training sequences, $n = 12$, filled) and high-depth (ca. ≥ 500, $n = 13$, hollow) MSAs. Arrows highlight the change from low to high depth; asterisks mark the per-metric depth-regime interaction test ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$; Appendix A.10.6). ProtGPT3-MSA's advantage over SFT-112M follows the same pattern as over SFT-10B: large for shallow MSAs and shrinking towards zero for deep ones. By contrast, scaling the SFT model from 112M to 10B parameters gives only small differences in either depth regime, with a significant interaction only for HH-score.

with 100 sequences to deeper ones approaching 2000, avoiding the bias toward high-depth MSAs that would result from naive uniform sampling over the full pool.

A.12 Aligned vs Unaligned conditional protein generation with ProtGPT3-MSA

ProtGPT3-MSA was trained in to main modalities 1) *aligned*: the 16 input sequences were passed to the model aligned including gap tokens (i.e., "−") and 2) *unaligned*: the 16 input sequences were passed in full (i.e., including insertions and without gaps). Therefore, the model has the ability of generating both *aligned* and *unaligned* protein sequences. When we compare few-shot prompting ProtGPT3-MSA with SFT of single-sequence ProtGPT3 models, we always used the *unaligned* modality to ensure a fair comparison (i.e., single-sequence models were not trained to process aligned sequences). Here, we investigate any potential difference in homolog generation quality between the two modalities. In particular, we look at potential difference in the quality of generated sequences at the level of MSA per-column statistics to assess any potential benefit of inputting already aligned sequences to the model. For this comparison, we randomly sampled 10 MSAs from the Protein Data Bank subset of OpenProteinSet (Ahdritz et al., 2023). From each MSA, we randomly sampled a set of 15 homologs and used it to (few-shot) prompt ProtGPT3-MSA in both *aligned* and *unaligned* mode. Next, we compared the quality of the generated sequence from each modality relative to the reference MSA using the positional KL-divergence and HMM profile comparison described in Appendix A.10. Fig. 11 shows there is not substantial difference in the quality of generated sequences between the two modalities.

Table 8: Characteristics of the 25 held-out test MSAs.

MSA ID	$n_{\text{sequences}}$	Neff[a]	Mean max fident (train–val)	Mean max TM-score (train–val)
0	119	80.1	0.743	0.523
1	139	118.4	0.731	0.284
2	143	81.7	0.792	0.508
3	169	148.0	0.602	0.377
4	171	127.0	0.617	0.611
5	194	167.8	0.526	0.679
6	222	82.1	0.797	0.807
7	225	196.4	0.568	0.898
8	229	182.9	0.595	0.862
9	293	218.0	0.661	0.840
10	392	332.9	0.525	0.801
11	399	338.7	0.531	0.823
12	506	376.5	0.698	0.810
13	515	389.8	0.629	0.869
14	548	511.4	0.430	0.863
15	967	815.3	0.646	0.851
16	1046	998.4	0.584	0.946
17	1050	1000.8	0.537	0.889
18	1073	998.4	0.575	0.911
19	1081	649.1	0.733	0.907
20	1121	1064.4	0.500	0.918
21	1243	1191.6	0.171	0.803
22	1578	1448.2	0.443	0.842
23	1702	1600.4	0.390	0.804
24	1827	1749.6	0.440	0.868

[a] Neff was computed as described by (Morcos et al., 2011) with re-weighting based on 80% sequence identity clustering.

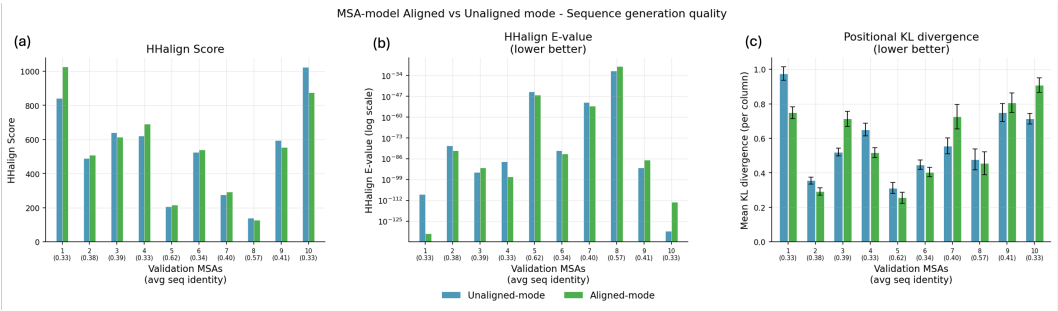

Figure 11: Comparison between sequences generated by ProtGPT3-MSA in *aligned* vs *unaligned* mode in terms of (a-b) HMM profile and (c) per-column KL-divergence relative to 10 reference MSAs. Within brackets on the x-axis, we report the average sequence identity for each reference MSA.

A.13 Prompt Ensembling Strategies for ProtGPT3-MSA

ProtGPT3-MSA can be prompted with up to 15 homologous sequences (i.e., pretrained on sets of 16 concatenated sequences). However, natural protein families typically contain hundreds to thousands of homologs, far exceeding this context budget. To leverage a larger fraction of the available evolutionary information during conditional generation, we introduce *prompt ensembling*: M independent context sets are sampled from the MSA pool (i.e., the target protein family) , the model scores each in parallel, and the resulting next-token distributions are combined into a single sampling distribution. We explore three strategies that differ in how this combination is performed, described below. Let M denote the number of ensemble members and $\mathcal{V}$ the vocabulary. At each generation step t, every ensemble member $m \in \{1, \ldots, M\}$ consists of a distinct set of homologous sequences $\mathcal{R}^{(m)}$ sampled from the target protein family (i.e., MSA). Given the current generation step $s_{<t}$

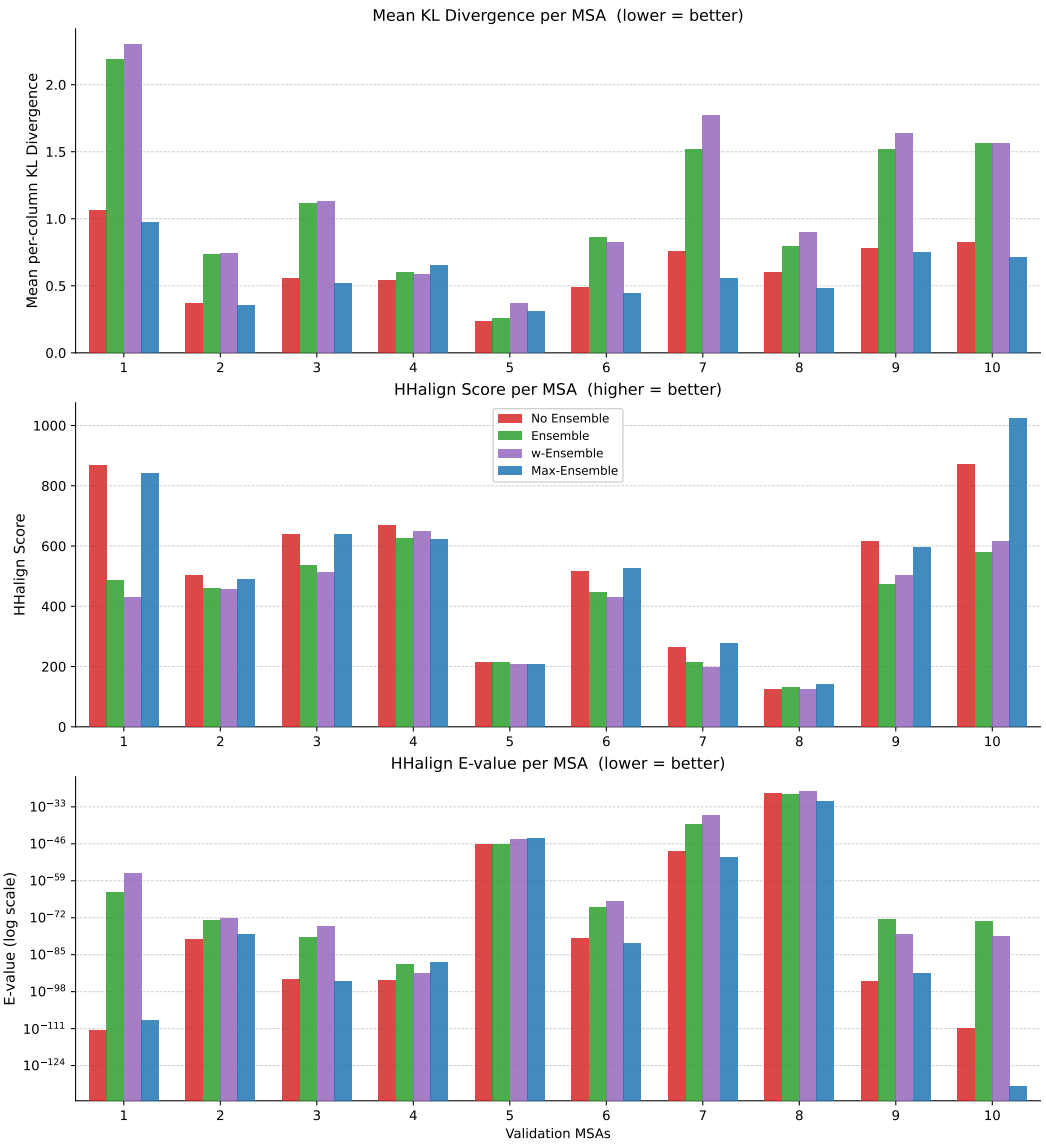

Figure 12: MSA model ensembling approaches comparison for 10 randomly sampled MSAs from the Protein Data Bank subset of OpenProteinSet (Ahdritz et al., 2023)

and context $\mathcal{R}^{(m)}$, the model produces a next-token log-probability vector $\ell_t^{(m)} = \log p_t^{(m)} \in \mathbb{R}^{|\mathcal{V}|}$, where $p_t^{(m)}$ is obtained by applying a softmax with temperature τ to the model logits.

Ensemble aggregates by averaging in log-space (equivalent to a geometric mean of probabilities) and renormalising:

$$\bar{p}_t \;\propto\; \exp\left(\frac{1}{M}\sum_{m=1}^{M} \ell_t^{(m)}\right). \tag{3}$$

where $\bar{p}_t$ denote the merged (unnormalised) next-token log-probability across the ensemble. The next token is sampled as $s_t \sim \bar{p}_t$ and appended to the shared prefix before the next decoding step. Tokens must be confidently predicted across *all* ensemble members to receive high probability mass, making this the most conservative of the three strategies.

w-Ensemble weights each member's distribution by the model's own confidence in its context. Specifically, a scalar quality score $q^{(m)}$ is assigned to each prompt by computing the mean per-token

log-probability over the prompt tokens $\mathcal{R}^{(m)}$ in a single forward pass prior to generation:

$$q^{(m)} = \frac{1}{|\mathcal{R}^{(m)}|} \sum_j \log p_\theta\left(r_j^{(m)} \mid r_{<j}^{(m)}\right). \tag{4}$$

These scores are converted to a weight distribution via softmax, $w^{(m)} = \mathrm{softmax}(\mathbf{q})_m$, and the ensemble distribution is the resulting weighted average in log-space:

$$\bar{p}_t \;\propto\; \exp\left(\sum_{m=1}^{M} w^{(m)}\, \ell_t^{(m)}\right). \tag{5}$$

The next token is sampled as $s_t \sim \bar{p}_t$ and appended to the shared prefix before the next decoding step. Prompts that the model assigns higher likelihood (i.e., suggesting better alignment with the learned sequence distribution) contribute proportionally more to the final prediction.

Max-Ensemble extends w-Ensemble by discarding all but the single most confident ensemble member. Using the same prompt scores $q^{(m)}$ from Eq. equation 4, the highest-scoring context set is selected as:

$$m^* \;=\; \arg\max_{m\in\{1,\ldots,M\}}\; q^{(m)}, \tag{6}$$

and the next token at each generation step t is sampled directly from that prompt's distribution:

$$s_t \;\sim\; p_\theta\left(\cdot \mid s_{<t},\, \mathcal{R}^{(m^*)}\right). \tag{7}$$

Because all non-selected prompts are pruned before the autoregressive loop begins, Max-Ensemble reduces wall-clock generation time by a factor of M relative to running a full M-member ensemble, while retaining the benefit of having scored a diverse pool of M candidate context sets.

In order to assess which of the 3 prompt ensemble approaches works best, we look at potential difference in the quality of generated sequences across the four approaches. In particular, we assess the quality of generated sequences in terms of MSA per-column statistics, which may better capture fine grained difference across the different approaches compared to more global sequence and structure similarity metrics. For this we randomly sampled 10 MSAs from the Protein Data Bank subset of OpenProteinSet (Ahdritz et al., 2023). From each MSA, we randomly sampled 10 sets of 15 homologs (i.e., 150 homologs in total) and used them to (few-shot) prompt ProtGPT3-MSA via one of the 3 ensemble approaches described above, 1) Ensemble, 2) w-Ensemble and 3) Max-Ensemble. Additionally, we also assess the performance of not using an ensemble approach (i.e., No Ensemble) by randomly sampling a single set of 15 homologs and use it to (few-shot) prompt ProtGPT3-MSA. We compared the quality of the generated sequence from the 4 approaches relative to the reference MSA using the positional KL-divergence and HMM profile comparison described in Appendix A.10. Fig. 12 shows Max-Ensemble performs best, albeit only marginally better than the No Ensemble approach. This finding implies that ProtGPT3-MSA own confidence (i.e., log-likelihood) can be used to pick the most informative sequences as prompt to improve generation. We find both w-Ensemble and Ensemble perform substantially worse than the two other approaches. We speculate this may be because both w-Ensemble and Ensemble rely on averaging over multiple predictions, which may dilute subtle evolutionary signals detected only by some of the prompts (Dinh et al., 2026).

A.14 Conditional generation comparison for defluorinases

Design and Computational Characterization ProtGPT3-MSA and single-sequence models were used for the targeted generation of defluorinases. Sequences with ProtGPT3-MSA were generated by prompting with the 7 homologous sequences reported in Probst et al. (Probst et al., 2025). We used the same 7 sequences to fine-tune the three single-sequence ProtGPT3 models. Given the very small number of sequences, we used k-fold cross validation to pick the optimal number of epochs to train each model. After finding the epoch achieving the lowest validation loss across the k-fold (for each model), we re-trained each model on all the 7 sequences for that optimal number of epochs. The learning rate was set to the same value as found in Appendix A.10.1. Next, we generated 3240 sequences per model (360 sequences for each combination of top-p $\in \{0.7, 0.9, 1.0\}$ and temperature $\in \{0.8, 0.9, 1.0\}$) and predicted their homodimeric structures with Boltz2 (Passaro et al., 2025) in single-sequence mode, while providing a template pdb file of an experimentally solved crystal structure from a defluorinase (PDB: 3UMG_A). All subsequent structural analyzes were performed

with the highest-ranking structure of the 5 generated predictions. Structural and sequential similarity comparisons against the reference protein structure and confirmed 7 defluorinase sequences were performed as explained above (see A.10). Additionally, we used PyRosetta (Chaudhury et al., 2010) to compute Rosetta Energy Scores (Leaver-Fay et al., 2011; Alford et al., 2017), and the ratio between the hydrophobic solvent accessible area of the protein complex and the idealized surface area for a protein complex of the same size as described by (Dill et al., 2011). For computational characterization, we define a broad success metric that classifies a design as successful if the generated sequence resembles a defluorinase sequentially (sequence identity to any known defluorinases between 85% and 95%), structurally (TM-Score ≥ 0.5) and passes a mild solubility filter (hydrophobic solvent accessible solvent area ratio ≤ 1.7). Given the unknown molecular features that separate defluorinases from dechlorinases, we reasoned that staying close in sequence space and structural space to experimentally confirmed defluorinases will be necessary to achieve reasonable experimental success rates in the absence of better filter criteria. Filtering for the hSASA ratio, is a high-level filter for protein solubility (Verkuil et al., 2022). We note that these filters can only be considered a minimum requirement for a successful design and do not reflect experimental success rates, which are expected to be lower. However, as only very few sequences generated by the fine-tuned ProtGPT3-1.3B and ProtGPT3-10B passed this initial set of filters, further filtering was only possible with sequences generated with ProtGPT3-MSA. Specifically, we ranked sequences by their Boltz2 pLDDT value (Passaro et al., 2025), spatial aggregation propensity, hSASA ratio, Rosetta energy score (Chaudhury et al., 2010; Leaver-Fay et al., 2011; Alford et al., 2017), interface pLDDT value, and the ligand-protein interface pTM score, hypothesizing that these would translate to higher experimental success rates beyond the initial broad filtering.

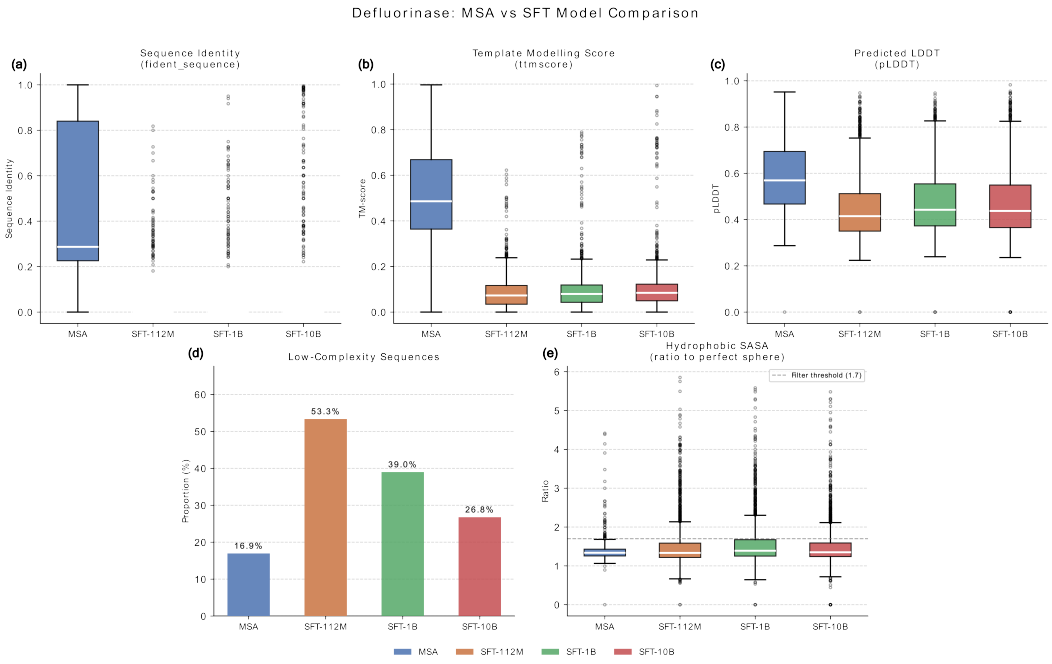

Figure 13: Computational characterization of defluorinases generated with ProtGPT3-MSA and single-sequence models prompted/fine-tuned with the available 7 sequences of experimentally confirmed defluorinases.

Cloning, expression and purification of dehalogenases The encoding sequences of the 61 candidates and the natural dehalogenases were purchased from GenScript Biotech® as double stranded DNA with flanking recognition sites for *Bsa*I at both 5'and 3' ends. Such sequences were cloned by Golden Gate in the pCoofy plasmid. *E. coli* DH5α cells were then transformed by heat-shock with the Golden Gate products and streaked onto LB-agar plates supplemented with kanamycin (50 μg/mL) and 1 mM isopropyl-β-D-1-thiogalactopyranoside (IPTG), presence of the genes of interest was confirmed by colony PCR and the expression vectors isolated using the GeneJET Plasmid Miniprep Kit from ThermoFisher Scientific® (Cat. K0502).

Protein expression of all sequences was carried out in *E. coli* BL21(DE3) cells, after transformation with the expression vectors by heat-shock, in 2YT auto-induction medium (auto2YT) purchased from Formedium™ (Cat. AIM2YT0260). Over-night (O/N) precultures were grown at 37ºC in 3 mL LB+kanamycin (50 μg/mL), then 40 μL were taken to inoculate 4 mL auto2YT+kanamycin (50 μg/mL)/construct. These cultures were incubated at 37ºC for 4 hours in 24-well plates with mild agitation, then transferred to 20ºC O/N.

Cells were harvested by centrifugation (3000 g, 20 minutes, 4ºC) and resuspended in 1 mL of BUFFER A (50 mM TRIS·H_2SO_4, 500 mM NaAcO, 20 mM imidazole), supplemented with 1X cOmplete™ protease inhibitor cocktail EDTA-free from Merck (Cat. 11873580001) and a few crystals of DNAse and lysozyme. Cell lysis was triggered by the addition of 1X FastBreak™ Cell Lysis Reagent from Promega®Corporation (Cat. V8571) and incubation at room temperature (r.t.) for 30 minutes with mild mixing, then lysates were centrifuged (3000 g, 1 h, 4ºC) and the supernatants recovered for batch purification with Pierce™ Ni-NTA Magnetic Agarose Beads (ThermoFisher Scientific®, Cat. 78606). The purification was carried out in an Opentrons Flex liquid-handling robot, in which samples were incubated for an hour with the beads. Then, we proceeded with removal of the supernatant, two wash steps with BUFFER B (50 mM TRIS·H_2SO_4, 500 mM NaAcO, 40 mM imidazole), and one single elution step with BUFFER C (50 mM TRIS·H_2SO_4, 500 mM NaAcO, 400 mM imidazole). Presence of the purified candidates and the natural sequences was checked by Bradford assay.

Defluorination activity assays of dehalogenases and fluoride detection The purified proteins were buffer exchanged to 50 mM TRIS·H_2SO_4, 500 mM NaAcO to remove the excess of imidazole using Zeba™ Spin Desalting Plates 7K MWCO from ThermoFisher Scientific® (Cat. 89808). Defluorination assays were performed in triplicates, 100 μL per replica, using 10 μg of each purified enzyme and fluoroacetic acid (FAcOH) at 1.25 mM as substrate; such reactions were incubated at 37ºC for 24 hours.

Defluorination reactions were stopped by thermal denaturation at 95ºC for five minutes. The inactivated samples were filtered with AcroPrep™ Advance 96-Well Filter Plates wwPTFE Membrane from Cytiva (Cat. 8582), and subjected to fluoride detection ((Probst et al., 2025). Absorbance measurements at 480 and 540 nm were recorded (Figs. 14, 15) in a TECAN plate-reader and ratios 480/540 calculated to compared fluoride release among all tested proteins (see Fig 2g in the main text).

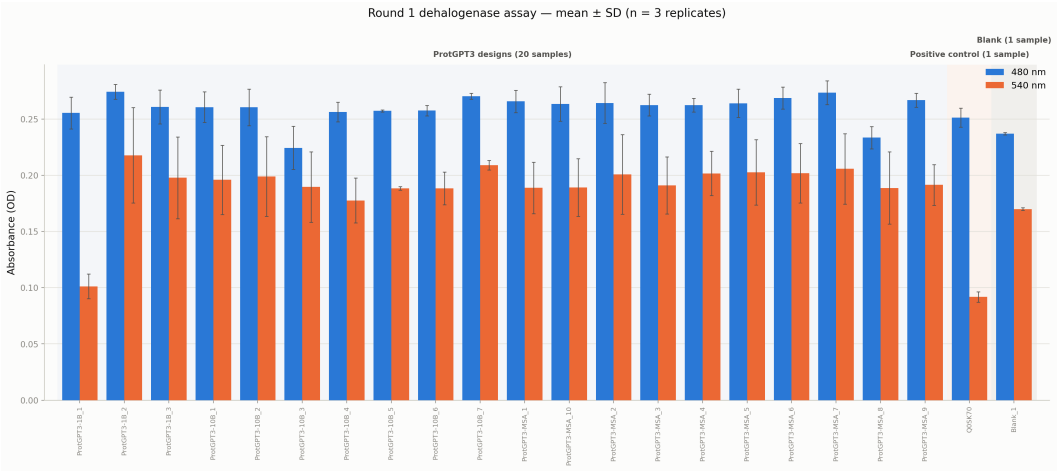

Figure 14: Comparison of the raw readings at 480 (blue) and 540 (orange) nm of the candidates designed with ProtGPT3-1.3B, ProtGPT3-10B, the first ten designs of ProtGPT3-MSA, and the dehalogenase from *Rhodococcus jostii* as positive control (Q0SK70). Each bar corresponds to the mean ± SD by triplicates of the absorbance values for each tested protein.

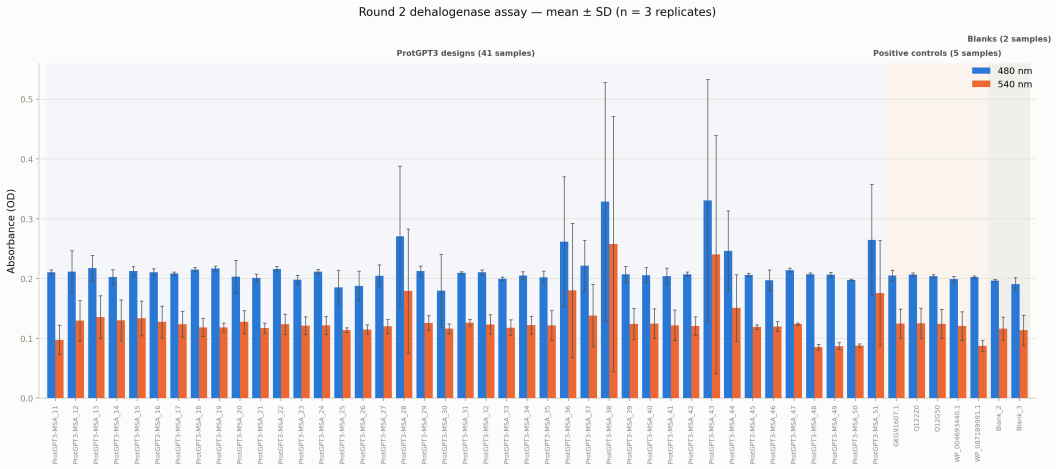

Figure 15: Comparison of the raw readings at 480 (blue) and 540 (orange) nm of the 41 other candidates designed with ProtGPT3-MSA and five additional natural sequences (GKG91607.1, Q122Z0, Q12G50, WP_004693440.1, WP_087189991.1). Each bar corresponds to the mean ± SD by triplicates of the absorbance values for each tested protein.

A.14.1 SFT Defluorinase Failure Analysis

The low SFT pass rates for the in-silico filters were not caused by low-temperature memorization and overfitting, as one may expect. No SFT-112M or SFT-1.3B generation exceeded 95% identity, and only a handful of SFT-10B generations did. Instead, most SFT generations drifted below 85% identity and failed to recover the target defluorinase fold. We swept sampling hyperparameters per model across the full $\{\text{top-}p, T\}$ grid (Table 9); the low pass rate persisted across configurations rather than being concentrated in any corner of the grid. Training hyperparameters (learning rate and epochs) were selected separately via k-fold cross-validation, as described in the main text. The low pass rate therefore reflects a genuine limitation of SFT sampling at this scale, not an undertrained or misconfigured model.

Table 9: Designs passing all filters across the sampling grid.

		Designs passing filters		
Model	top-p	$T = 0.8$	$T = 0.9$	$T = 1.0$
MSA	0.7	83	93	76
MSA	0.9	64	49	27
MSA	1.0	36	13	3
SFT-112M	0.7	0	0	0
SFT-112M	0.9	0	0	0
SFT-112M	1.0	0	0	0
SFT-1.3B	0.7	0	2	0
SFT-1.3B	0.9	0	0	1
SFT-1.3B	1.0	0	0	0
SFT-10B	0.7	2	0	1
SFT-10B	0.9	1	0	1
SFT-10B	1.0	0	1	1

To identify the dominant failure mode, we removed the sequence-identity criterion and retained only the structural and solubility filters. Over 99% of SFT designs still failed (Table 10), showing that structural similarity to the target fold is an equally decisive limitation.

Table 10: Filter pass rates and structural similarity to the target fold.

Model	Passed all filters	Passed ignoring identity	Median TM-score
MSA	444 (13.70%)	1,510 (46.60%)	0.486
SFT-112M	0 (0.00%)	7 (0.22%)	0.073
SFT-1.3B	3 (0.09%)	24 (0.74%)	0.079
SFT-10B	7 (0.22%)	32 (0.99%)	0.084

Finally, the SFT models rarely produce sequences with more than 95% identity to the prompt at any hyperparameter setting; only SFT-10B does so, and only for a few designs (Table 11). This rules out low-temperature memorisation as an explanation for the SFT results.

Table 11: Designs with sequence identity $> 95\%$ to the prompt, by temperature.

Model	$T = 0.8$	$T = 0.9$	$T = 1.0$
MSA	179	106	54
SFT-112M	0	0	0
SFT-1.3B	0	0	0
SFT-10B	6	8	8

Table 12: Active generated sequences detected in the defluorinase activity assay.

ID	**Model**	**Sequence**
SFT-1.3B_1	SFT-1.3B	MLEAIAFDAYGTLYDVYSVERKCEEEYPGSGAAMSRLWRQKQLEYSWLRTLMGRYADFWR VTEDALRYTLAELGLEGDGGKIAAIMEGYLELEMYPEVIEAMQRFGPRKKAILTNGNLGM IEPLVARSGLDGHLDACLSADEAGLFKTRPEVYQLAVDHLGVARERLLFVSSNGWDVAGA KAFGFTVGWLNRRGLPPEELGVRADYEASDLLELTERLAAEEG
MSA_48	MSA	MNRTKIVALDLFGTLVDWQSAVAQEAKSLALNIDANQFALDWRAGYRPAMDQVLSGQQPW TSIDDIHRLILDELLAKYQISTLTEAQKMDLNFIWHRLNPWPDTVAALNQLKQDYIICTL SNGNIRLLVDLAKYAKLPWDTIFSAENFKAYKLSPKTYLGVSDFLNVAPSQVMMVATHQD DLAAARGCGLRTAYIERPFEYGAAQLKDSSPCIDNNLHATDLLNLVSLLKEKA
MSA_49	MSA	MLTLISFDTFGTLFDFYGTLVEVESAEALRSRLFPGQGEAVGRLWRDKQLEYTWLRTLMG RYADFWRVTEDALRYTLAELGLEGDEGKIAAIMATYLELEMYPEVIEAMARFGTRKKAIL TNGNLGMIQPLVARSGLGAHLDACLSADEAGLFKVRPEVYQLAVDHLGVARERLLFVSSN GWDVAGAKAFGFTVGWLNRRGLPPEELGVRADYEASNLLELAEMVVSA
MSA_50	MSA	MSISSVRIVIFDLYGTLVDYESGVAREIKEIAFDIDANQFALDWRAGYRPAMDQVLSGQQ PWTSIDDIHRLILDELLAKYQISTLTEAQKMDLNFIWHRLNPWPDTVAALNQLKQDYIIC TLSNGNIRLLVDLAKYAKLPWDTIFSAENFKAYKLSPKTYLGVSDFLNVAPSQVMMVATH QDDLAAARGCGLRTAYIERPFEYGAAQLKDSSPCIDNNLHATDLLNLVSLLKEKA

A.15 FK-SMC SAMPLING WITH PROTGPT3-MSA

Algorithm 1: FK-SMC Sampling with ProtGPT3-MSA

Initialise prompt $P \leftarrow P_0$
Initialise model $\mathcal{M}_{\text{ProtGPT3-MSA}}$
Initialise scoring function $s(\cdot)$
Initialise number of parallel trajectories N
Initialise number of homolog rollout steps K
Initialise maximum prompt size $T = 16$

while $|P| < T$ **do**
  for $k = 1, \ldots, K$ **do**
    Sample N sequences in parallel:
    $x_k^{(n)} \sim \mathcal{M}_{\text{ProtGPT3-MSA}}\left(\cdot \mid P \oplus x_1^{(n)} \oplus \cdots \oplus x_{k-1}^{(n)}\right), \quad n = 1, \ldots, N$
  end for
  Score each trajectory $\tau_n = \left(x_1^{(n)}, \ldots, x_K^{(n)}\right)$:
  $S(\tau_n) = \frac{1}{K} \sum_{k=1}^{K} s\left(x_k^{(n)}\right), \quad n = 1, \ldots, N$
  Select best trajectory: $n^* = \arg\max_n S(\tau_n)$
  Append to prompt: $P \leftarrow P \oplus x_1^{(n^*)}$
end while
return P

We adapt a Feynman–Kac-style sequential Monte Carlo (FK-SMC) procedure to steer sequence generation from ProtGPT3-MSA toward a target objective, $s(\cdot)$. The procedure operates iteratively: at each step, the model is prompted with a growing sequence of homologs and branches into multiple candidate trajectories, which are scored and selectively retained to bias the next generation step. Formally, let P_0 denote an initial prompt of one or more concatenated homolog sequences, N the number of candidate trajectories sampled per step, and K the rollout length (number of homolog sequences per trajectory). At each step t, we sample N trajectories from ProtGPT3-MSA conditioned on the current prompt P_{t-1}. Each trajectory $\tau_n = (x_1^{(n)}, \cdots, x_1 K(n)$ consists of K homolog sequences generated autoregressively. Trajectories are scored by averaging a sequence-level objective $s(\cdot)$ across all K sequences. The highest-scoring trajectory is selected, its first sequence is appended to the prompt, and the process repeats. Both N and K are controllable hyperparameters governing the breadth and depth of the search. In practice, we terminate the generation when the prompt reaches 16 sequences (i.e., the max n. of sequences that ProtGPT3-MSA can process), although the search could continue indefinitely by simply starting to remove the first sequence in the prompt at each new step. The procedure is summarised in Algorithm 1.

A.15.1 FK-SMC STEERING FOR GB1 FITNESS

To test whether FK-steering's compute-efficiency advantage generalizes beyond protein stability and to a higher number of mutations, we apply the same FK-SMC steering procedure to GB1, the IgG-binding B1 domain of protein G, using Wu et al. (2016) DMS: 149,361 measured variants covering 93.4% of the full 20^4 combinatorial space at four co-varying positions (39, 40, 41, and 54 of the wild-type sequence). We restrict to the 121,174 variants carrying exactly four substitutions (quadruple mutants). As for stability, we train a GRPO oracle using ProtGPT3-112M ($\beta = 0.1$, learning rate 1×10^{-4}) on a percentile-balanced resample of the quadruple-mutant training split so its own sequence log-likelihood tracks fitness. The trained model reaches a Spearman correlation of $\rho = 0.474$ on the training split and $\rho = 0.454$ on the held-out validation split between oracle log-likelihood and true fitness (see Fig. 16). For each of 40 random seeds, we sample 6 variants uniformly from the bottom 25th fitness percentile as the initial prompt for FK-SMC, then compare FK-steering against a best-of-N baseline at a matched compute budget of 900 sequences generated and oracle-scored per seed.

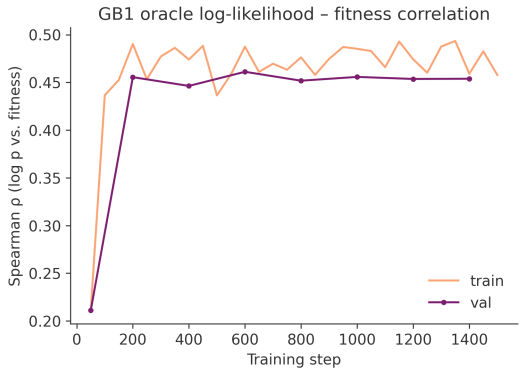

Figure 16: Spearman correlation between the oracle's sequence log-likelihood and ground-truth GB1 protein fitness (Wu et al., 2016), on the training (orange) and validation (purple) splits, across GRPO training steps.

A.16 Single-sequence model alignment to high complexity and high pLDDT

To improve the quality of generated sequences across the 3 sizes of single-sequence models, we generate 80 thousands sequences across all combinations of top-p $0.6, 0.8, 0.9, 1.0$ and temperature $0.6, 0.8, 0.9, 1.0$ for each model. Next, we score all generated sequences based on $\%$ of low complexity regions as well as pLDDT (predicted by ESMFold (Lin et al., 2023)) and classify them as "pass", each sequence that has plDDT > 0.7 as well as percentage of low complexity regions $< 25\%$, while classifying all other sequences as "fail". Next, we cluster "pass" and "fail" sequence separately at 50% sequence identity and 0.8 coverage using MMseqs2 (Steinegger & Söding, 2017). We take a single sequence form each cluster and use them to build a binary preference dataset. Since the ratio of "fail" sequence was much higher than the "pass" one, especially for the smaller models, we build the preference dataset by sampling 20 "fail" sequences for each "pass" sequence. Importantly, we enforce a strict sequence length match between pairs of "pass" and "fail" sequences to avoid the model learning to distinguish the two in terms of sequence length instead of complexity and pLDDT. Based on this preference dataset, we use the Direct Preference Optimization (DPO) (Rafailov et al., 2023) algorithm to shift each model distribution towards high complexity and high pLDDT sequences. We repeat this entire process for 3 iteration, apart from ProtGPT3-10B, which after a single iteration achieved a 5% proportion of low complexity sequences.

A.16.1 Intrinsic Reward Discrimination Analysis

DPO alignment was performed using model-generated sequences scored against pLDDT and LCR thresholds; here we test whether the aligned models have internalised these objectives when evaluated on real, naturally occurring protein sequences. We randomly subsample sequences from the PDB, which include AlphaFold-predicted pLDDT scores, and compute the percentage of low-complexity regions (LCR) for each sequence. Sequences are partitioned into two groups using the thresholds applied during DPO training: *low-quality* sequences have pLDDT < 70 and percentage of low complexity regions $> 25\%$, and *high-quality* sequences have pLDDT ≥ 70 and percentage of low complexity regions $< 25\%$, with groups balanced by random subsampling of the majority class. For each aligned model (112M, 1.3B, 10B parameters), we compute the intrinsic reward of each sequence relative to its corresponding base model,

$$r(x) = \beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)} \tag{8}$$

where π_θ is the aligned model, π_{ref} is the base (reference) model and β is the KL penalty coefficient use in DPO. We report the AUC-ROC of $r(x)$ as a binary classifier between the two groups, with a t-test for significance (see Fig. 17d). AUC values in the range 0.7–0.8 across model sizes indicate that the aligned models assign systematically higher intrinsic rewards to high-quality sequences, demonstrating generalisation of the learned preference to real protein sequences not seen during alignment. Importantly, we do not expect nor desire a near-perfect AUC: low-pLDDT and high-complexity sequences are naturally occurring and present in the pretraining corpus, so both π_θ

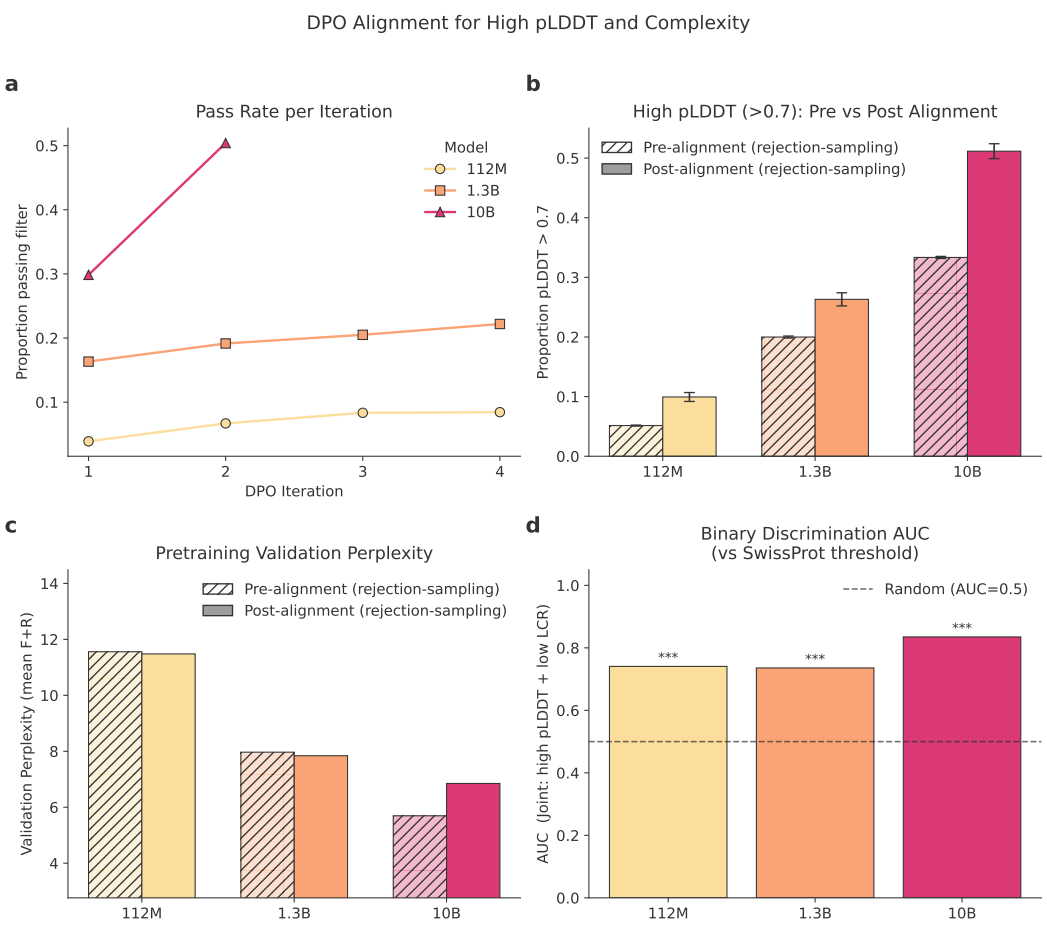

Figure 17: a) Pass rate denotes the proportion of generated sequences with plDDT > 0.7 as well as percentage of low complexity regions $< 25\%$, which grows at each iteration of the DPO alignment pipeline. b) Proportion of sequence with plDDT > 0.7 before and after alignment. c) Model perplexity on the held-out pretraining validation sequences before and after alignment, which can be used as a proxy for model forgetting pretraining during the DPO alignment pipeline. d) AUC-ROC for each model intrinsic reward based on true sequences randomly sampled from the PDB dataset.

and π_{ref} assign non-trivial likelihood to them. Since $r(x)$ is a relative quantity, a randomly drawn low-quality sequence may still receive a higher intrinsic reward than a randomly drawn high-quality sequence if it is intrinsically more likely under the base model (e.g., short or repetitive sequences that are statistically common). A perfect AUC would indicate the model has learned to categorically reject an entire class of naturally occurring proteins, which would reflect overfitting to the alignment objective rather than a balanced internalisation of structural preferences.

A.17 Hyperparameters

We report the main hyperparameters across the 4 models in the tables below, all other hyperparameters can be found in the model configuration and code at https://huggingface.co/protgpt3and `https://anonymous.4open.science/r/protGPT3-2053`.

Table 13: Hyperparameters used to train the ProtGPT3-MSA

Hyperparameter	Value
Max sequence length	16,384
Router aux. loss coef.	0.05
Optimizer	AdamW
Learning rate	2×10^{-4}
(β_1, β_2)	(0.9, 0.95)
Weight decay	0.1
Gradient clipping	1.0
Grad. accum. steps	16
Max tokens per batch	100,000
Precision	bfloat16
N. training GPUs	4

Table 14: Hyperparameters used to train the ProtGPT3-112M

Hyperparameter	Value
Max sequence length	1024
Router aux. loss coef.	0.05
Optimizer	AdamW
Learning rate	5×10^{-4}
(β_1, β_2)	(0.9, 0.999)
Weight decay	0.1
Gradient clipping	1.0
Grad. accum. steps	1
Batch size	500
Precision	bfloat16
N. training GPUs	4

Table 15: Hyperparameters used to train the ProtGPT3-1.3B

Hyperparameter	Value
Max sequence length	1024
Router aux. loss coef.	0.05
Optimizer	AdamW
Learning rate	2.5×10^{-4}
(β_1, β_2)	(0.9, 0.999)
Weight decay	0.1
Gradient clipping	1.0
Grad. accum. steps	4
Batch size	100
Precision	bfloat16
N. training GPUs	16

Table 16: Hyperparameters used to train the ProtGPT3-10B

Hyperparameter	Value
Max sequence length	1024
Router aux. loss coef.	0.05
Optimizer	AdamW
Learning rate	1×10^{-4}
(β_1, β_2)	(0.9, 0.999)
Weight decay	0.1
Gradient clipping	1.0
Grad. accum. steps	1
Batch size	300
Precision	bfloat16
N. training GPUs	40

A.18 Infrastructure and optimization

All models were trained on NVIDIA H100 GPUs, using FlashAttention-2 (Dao, 2023), online mini-batch packing (Kundu et al., 2024) as well as Liger Kernel (Hsu et al., 2025) for maximal efficiency. Additionally, we used the DeepSpeed Library (Rasley et al., 2020) for efficient distributed data parallel (DDP) across all models as well as for ZeRO-3 (Rajbhandari et al., 2020) for training, SFT and aligning the 10B model.

A.19 Models and code

All 7 models are publicly available open source on Hugging Face Hub at https://huggingface.co/protgpt3under a pseudonym to preserve anonymity. The entire code base for training and evaluation is available on a GitHub repository at `https://anonymous.4open.science/r/protGPT3-2053/README.md`.