Protein generation with evolutionary diffusion: sequence is all you need

Sarah Alamdari[1], Nitya Thakkar[2,†], Rianne van den Berg[3], Alex X. Lu[1], Nicolo Fusi[1], Ava P. Amini[1], Kevin K. Yang[1,*]

[1]Microsoft Research, Cambridge, MA, USA

[2]Brown University, Providence, RI, USA

[3]Microsoft Research AI4Science, Amsterdam, Netherlands

[†]Work done principally during an internship at Microsoft Research

[*]To whom correspondence should be addressed; E-mail: yang.kevin@microsoft.com.

Abstract Deep generative models are increasingly powerful tools for the *in silico* design of novel proteins. Recently, a family of generative models called diffusion models has demonstrated the ability to generate biologically plausible proteins that are dissimilar to any actual proteins seen in nature, enabling unprecedented capability and control in *de novo* protein design. However, current state-of-the-art models generate protein structures, which limits the scope of their training data and restricts generations to a small and biased subset of protein design space. Here, we introduce a general-purpose diffusion framework, EvoDiff, that combines evolutionary-scale data with the distinct conditioning capabilities of diffusion models for controllable protein generation in sequence space. EvoDiff generates high-fidelity, diverse, and structurally-plausible proteins that cover natural sequence and functional space. Critically, EvoDiff can generate proteins inaccessible to structure-based models, such as those with disordered regions, while maintaining the ability to design scaffolds for functional structural motifs, demonstrating the universality of our sequence-based formulation. We envision that EvoDiff will expand capabilities in protein engineering beyond the structure-function paradigm toward programmable, sequence-first design.

Evolution has yielded a diversity of functional proteins that precisely modulate cellular processes. Recent years have seen the emergence of deep generative models that aim to learn from this diversity to generate proteins that are both valid and novel, with the ultimate goal of then tailoring function to solve outstanding modern-day challenges, such as the rapid development of targeted therapeutics and vaccines or engineered enzymes for the degradation of industrial waste (**Fig. 1A**) (*1, 2*). Diffusion models provide a particularly powerful framework for generative modeling of novel proteins, as they generate high-diversity samples and can be conditioned given a wide variety of inputs or design objectives (*3–6*). Indeed, today's most biologically-plausible instances of *in silico*-designed proteins come from diffusion models of protein *structure* (*7–15*).

These models – including the current state-of-the-art approach RFdiffusion (*10*) – fit in the structure-based protein design paradigm of first generating a structure that fulfills desired constraints and then designing a sequence that will fold to that structure. However, sequence, not structure, is the universal design space for proteins. Every protein is completely defined by its amino-acid sequence. We discover proteins by finding their coding sequences in genomes, and proteins are synthesized as amino-acid sequences. Sequence then determines function through both an ensemble of structural conformations and the chemistry enabled by the amino acids themselves. However, not every protein folds into a static structure. In these cases, structure-based design is not viable because the function is not mediated by a static structure (*16–18*), with the most extreme examples being intrinsically disordered regions (IDRs) (*19*). Therefore, static structures characterized by X-ray crystallography are an incomplete distillation of the information captured in sequence space (*20–23*). Furthermore, structural data (ca. 200k solved structures in PDB) is scarce and unrepresentative of the full diversity of natural sequences (ca. billions of unique natural protein sequences; **Fig. 1A**), inherently limiting the capacity of any structure-based generative model to learn the full diversity of protein functional space.

We combine evolutionary-scale datasets with diffusion models to develop a powerful new generative modeling framework, which we term EvoDiff, for controllable protein design from sequence data alone (**Fig. 1**). Given the natural framing of proteins as sequences of discrete tokens over an amino acid language, we use a *discrete* diffusion framework in which a forward process iteratively corrupts a protein sequence by changing its amino acid identities, and a learned reverse process, parameterized by a neural network, predicts the changes made at each iteration (**Fig. 1B**). The reverse process can then be used to generate new protein sequences starting from random noise (**Fig. 1C**). Importantly, EvoDiff's discrete diffusion formulation is mathematically distinct from continuous diffusion formulations previously used for protein structure design (*7–15*). Beyond evolutionary-scale datasets of single protein sequences, multiple sequence alignments (MSAs) inherently capture evolutionary relationships by revealing patterns of conservation and variation in the amino acid sequences of sets of related proteins. We thus additionally build discrete diffusion models trained on MSAs to leverage this additional layer of evolutionary information to generate new single sequences (**Fig. 1C-D**).

We evaluate our sequence and MSA models – EvoDiff-Seq and EvoDiff-MSA, respectively – across a range of generation tasks to demonstrate their power for controllable protein design (**Fig. 1D**). We first show that EvoDiff-Seq unconditionally generates high-quality, diverse proteins that capture the natural distribution of protein sequence, structural, and functional space. Using EvoDiff-MSA, we achieve evolution-guided design of novel sequences conditioned on an alignment of evolutionarily-related, but distinct, proteins. Finally, by exploiting the conditioning capabilities of our diffusion-based modeling framework and its grounding in a universal design space, we demonstrate that EvoDiff can reliably generate proteins with IDRs, directly overcoming a key limitation of structure-based generative models, and generate scaffolds for functional structural motifs without any explicit structural information.

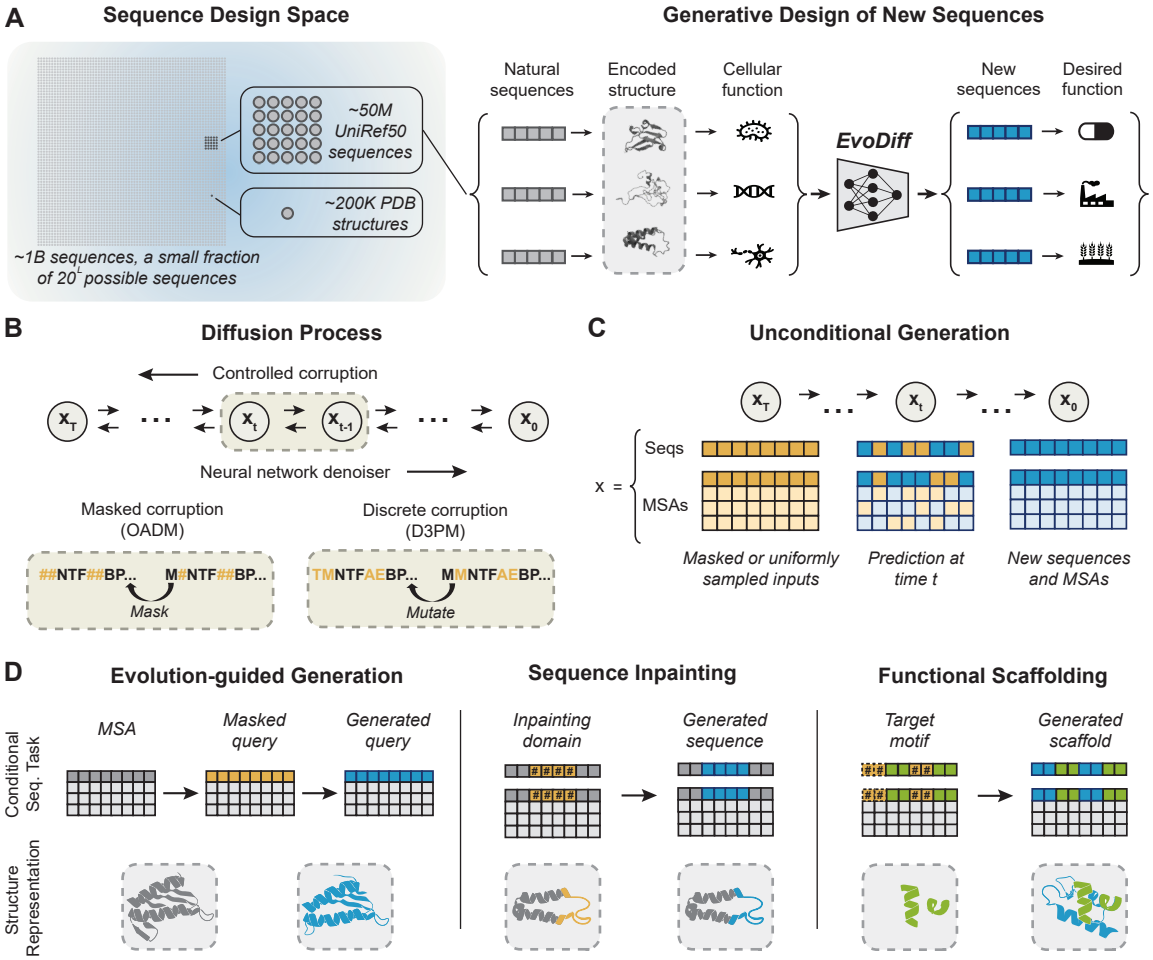

Figure 1: **Protein sequence generation with evolutionary diffusion.** **(A)** (Left) Evolution has sampled a tiny fraction of possible protein sequences. Experimental structures have been determined for even fewer proteins. (Right) EvoDiff is a generative discrete diffusion model trained on natural protein sequences. Sampling from EvoDiff yields new protein sequences that may perform desired functions. **(B)** Discrete diffusion models consist of controlled corruption and learned denoising processes. In the masked corruption process, input tokens are masked in an order-agnostic fashion (bottom, left). In the discrete corruption process, inputs are corrupted via a Markov process controlled by a transition matrix capturing amino acid mutation frequencies (bottom, right). **(C)** EvoDiff enables unconditional generation of protein sequences or MSAs. Starting from masked or uniformly sampled inputs x_T, EvoDiff generates new sequences or MSAs by reversing the corruption process, iteratively denoising x_t into realistic sequences or MSAs x_0. **(D)** Controllable protein design with EvoDiff, via conditioning on evolutionary information encoded in MSAs (left); inpainting functional domains from masked portions of a sequence (middle); or scaffolding structural motifs without explicit structural information (right).

Discrete diffusion models of protein sequence EvoDiff is the first generative diffusion model for protein design trained on evolutionary-scale protein sequence data. We investigated two types of forward processes for diffusion over discrete data modalities (*24, 25*) to determine which would be most effective (**Fig. 1B**). In order-agnostic autoregressive diffusion (EvoDiff-OADM, see Methods) (*24*), one amino acid is converted to a special mask token at each step in the forward process (**Fig. 1B**). After T=L steps, where L is the length of the sequence, the entire sequence is masked. We additionally designed discrete denoising diffusion probabilistic models (EvoDiff-D3PM, see Methods) (*25*) for protein sequences. In EvoDiff-D3PM, the forward process corrupts sequences by sampling mutations according to a transition matrix, such that after T steps the sequence is indistinguishable from a uniform sample over the amino acids (**Fig. 1B**). In the reverse process for both, a neural network model is trained to undo the previous corruption. The trained model can then generate new sequences starting from sequences of masked tokens or of uniformly-sampled amino acids for EvoDiff-OADM or EvoDiff-D3PM, respectively (**Fig. 1C**).

To facilitate direct and quantitative model comparisons, we trained all EvoDiff sequence models on 42M sequences from UniRef50 (*26*) using a dilated convolutional neural network architecture introduced in the CARP protein masked language model (*27*). We trained 38M-parameter and 640M-parameter versions for each forward corruption scheme to test the effect of model size on model performance. As a first evaluation of our EvoDiff sequence models, we calculated each model's test-set perplexity, which reflects its ability to capture the distribution of natural sequences and generalize to unseen sequences (see Methods). We observe that EvoDiff-OADM learns to reconstruct the test set more accurately than two tested EvoDiff-D3PM variants employing uniform and BLOSUM62-based transition matrices (**Table S1; Fig. S1**). Furthermore, EvoDiff-OADM is the only model variant where performance scales with increased model size (**Table S1; Fig. S1**).

To explicitly leverage evolutionary information, we designed and trained EvoDiff MSA models using the MSA Transformer (*28*) architecture on the OpenFold dataset (*29*). To do so, we subsampled MSAs to a length of 512 residues per sequence and a depth of 64 sequences, either by randomly sampling the sequences ("Random") or by greedily maximizing for sequence diversity ("Max"). Within each subsampling strategy, we then trained EvoDiff MSA models with the OADM and D3PM corruption schemes. OADM corruption results in the lowest validation set perplexities, indicating that OADM models are best able to generalize to new MSAs (**Table S2; Fig. S2**). To select a subsampling method, we compared the ability of each model to reconstruct validation set MSAs, finding that maximizing for sequence diversity yields improved performance no matter how the validation MSAs are subsampled (**Table S2**). We thus selected the OADM-Max model for downstream analysis, hereafter referring to it as EvoDiff-MSA.

Structural plausibility of generated sequences We next investigated whether EvoDiff could generate new protein sequences that were individually valid and structurally plausible. To assess this, we developed a workflow that evaluates the foldability and self-consistency of sequences generated by EvoDiff (**Fig. 2A**). We generated 1000 sequences from each EvoDiff sequence model with lengths drawn from the empirical distribution of lengths in the training set. We compared EvoDiff's generations to sequences generated from a left-to-right autoregressive language model (LRAR) with the same architecture and training set as EvoDiff and to sequences generated from protein masked language models such as ESM-2 (*30*) (**Figs. 2B-C, S3, S4; Table S3**).

We assessed the foldability of individual sequences by predicting their corresponding structures using OmegaFold (*31*) and computing the average predicted local distance difference test (pLDDT) across the whole structure (**Fig. 2B**). pLDDT reflects OmegaFold's confidence in its

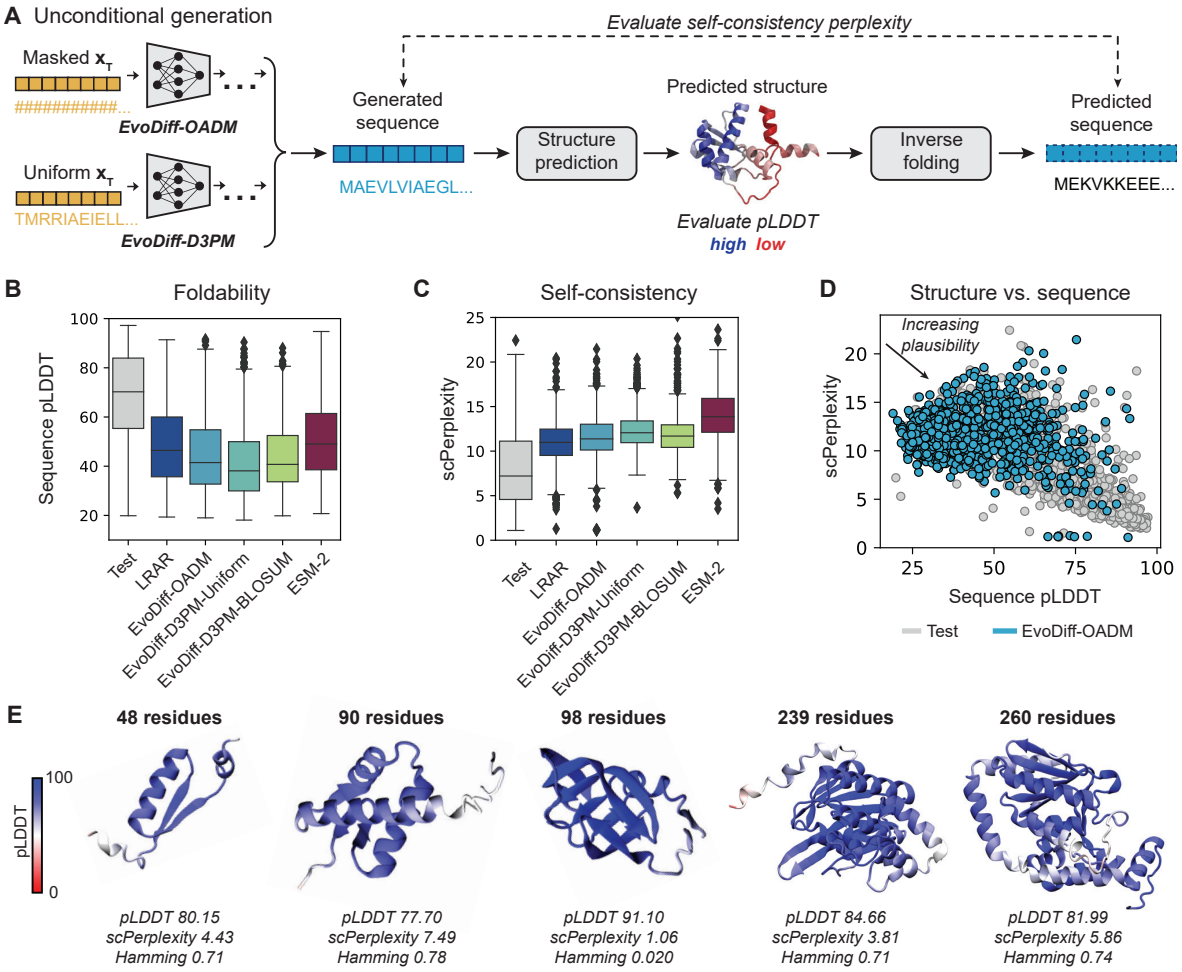

Figure 2: **EvoDiff generates realistic and structurally-plausible protein sequences**. **(A)** Workflow for evaluating the foldability and self-consistency of sequences generated by EvoDiff sequence models. **(B-C)** Distributions of foldability, measured by sequence pLDDT of predicted structures (B), and self-consistency, measured by scPerplexity (C), for sequences from the test set, EvoDiff models, and baselines (n=1000 sequences per model; box plots show median and interquartile range). **(D)** Sequence pLDDT versus scPerplexity for sequences from the test set (grey, n=1000) and the 640M-parameter OADM model EvoDiff-Seq (blue, n=1000). **(E)** Predicted structures and metrics for representative structurally plausible generations from EvoDiff-Seq, the 640M-parameter OADM model.

structure prediction for each residue. In addition to the average pLDDT across a whole protein, we observe that pLDDT scores can vary significantly across a protein sequence (**Fig. S5**). It is important to note that while pLDDT scores above 70 are often considered to indicate high prediction confidence, low pLDDT scores can be consistent with intrinsically disordered regions (IDRs) of proteins (*32*), which are found in many natural proteins. As an additional metric of structural plausibility, we computed a self-consistency perplexity (scPerplexity) by redesigning each predicted structure with the inverse folding algorithm ESM-IF (*33*) and computing the perplexity against the original generated sequence (**Fig. 2A, C; Table S3**). Given that ESM-IF and EvoDiff were both trained on UniRef50 data, it is possible that sequences from EvoDiff's validation set overlap with sequences in the ESM-IF train set; thus we performed the same self-consistency evaluations using ProteinMPNN (*34*), which is not trained on UniRef50, for inverse folding (**Table S3**).

While no generative model approaches the test set values for foldability and self-consistency, EvoDiff-OADM outperforms EvoDiff-D3PM and improves when increasing the model size (**Fig. 2B-D**; **Table S3**). We therefore selected the 640M-parameter EvoDiff-OADM model for downstream analysis and hereafter refer to it as EvoDiff-Seq. While a left-to-right autoregressive (LRAR) protein language model generates slightly more structurally-plausible sequences (**Table S3**), EvoDiff-Seq offers the advantage of direct, flexible conditional generation due to its order-agnostic decoding. Unconditional generation from masked language models produces less structurally-plausible sequences because of the mismatch between the training and generation tasks (**Table S3**). Analysis of representative examples of structurally plausible sequences sampled from EvoDiff-Seq across 4 different sequence lengths illustrates their structural plausibility and novelty from sequences in the training set, demonstrating that EvoDiff generates protein sequences that are individually valid (**Fig. 2E**).

Biological properties of generated sequence distributions Having shown that EvoDiff's generations are individually foldable and self-consistent, we next evaluated how well the *distribution* of designed protein sequences covered natural protein space. Ideally, generated sequences should capture the natural distribution of sequence, structural, and functional properties while still being diverse from each other and from natural sequences.

Previous work has shown that even without explicit supervision, protein language model embeddings contain information about both sequence and function as captured in GO annotations (*35, 36*). To evaluate coverage over the distribution of sequence and functional properties, we embedded each generated sequence using ProtT5 (*37*), a protein language model explicitly benchmarked for imputing GO annotations (*35*), and calculated the embedding space Fréchet distance between a set of generated sequences and the test set, where lower distance reflects better coverage. We refer to this metric as the Fréchet ProtT5 distance (FPD) and visualize these embeddings and the corresponding FPDs for sequences generated by EvoDiff-Seq and baseline models (**Figs. 3A, S6, S7; Table S1**). For RFdiffusion, we unconditionally generated 1000 structures with the same lengths as for EvoDiff-Seq and then used ESM-IF (*33*) to design their sequences. Both qualitatively and quantitatively, EvoDiff-Seq generates proteins that better recapitulate natural sequence and functional diversity than sampling from a state-of-the-art protein masked language model (ESM-2) or predicting sequences from structures generated by a state-of-the-art structure diffusion model (RFdiffusion) (**Fig. 3A**).

To evaluate the distribution of structural properties in generated sequences, we computed 3-state secondary structures (*38*) for each residue in generated and natural sequences and quantitatively compared the resulting distributions of structural properties to the distribution for the test set (**Figs. 3B, S8**). EvoDiff-Seq generates proportions of strands and disordered regions that are much more similar to those in natural sequences, while ESM-2 and RFdiffusion both generate proteins enriched in helices (**Fig. 3B**). To ensure our models were not memorizing training

data, we calculated the Hamming distance between each generated sequence and all training sequences of the same length and reported the minimum Hamming distance, representing the closest match of any generated sequence to any sequence in the train set (**Table S1**). On average, a sequence generated from EvoDiff-Seq has a Hamming distance of 0.83 from the most similar training distance of the same length. Together, these results demonstrate, via comparison to ESM-2 and RFdiffusion, that EvoDiff's diffusion objective and evolutionary-scale training data are both necessary to generate novel sequences that cover protein sequence, functional, and structural space.

Conditional sequence generation for controllable design EvoDiff's OADM diffusion framework induces a natural method for conditional sequence generation by fixing some subsequences and inpainting the remainder. Because the model is trained to generate proteins with an arbitrary decoding order, this is easily accomplished by simply masking and decoding the desired portions. We applied EvoDiff's power for controllable protein design across three scenarios: conditioning on evolutionary information encoded in MSAs, inpainting functional domains, and scaffolding functional structural motifs (**Fig. 1D**).

Evolution-guided protein generation with EvoDiff-MSA First, we tested the ability of EvoDiff-MSA to generate query sequences conditioned on the remainder of an MSA, thus generating new members of a protein family without needing to train family-specific generative models. We masked the query sequences from 250 randomly-chosen MSAs from the validation set and newly generated these sequences using EvoDiff-MSA. We then evaluated the quality of the resulting conditionally-generated query sequences via our foldability and self-consistency pipeline (**Fig. 4A**). We find that EvoDiff-MSA generates more foldable and self-consistent sequences than sampling from ESM-MSA (*28*) or using Potts models (*39*) trained on individual MSAs (**Figs. 4B-C, S9; Table S4**). To evaluate sample diversity, we computed the aligned

residue-wise sequence similarity between the generated query sequence and the most similar sequence in the original MSA. In contrast to sampling from a Potts model, generating from EvoDiff-MSA yields sequences that exhibit strikingly low similarity to those in the original MSA (**Fig. 4D; Table S4**) while still retaining structural integrity relative to the original query sequences (**Fig. 4E-F; Table S4**). To showcase these properties, we visualize OmegaFold-predicted structures and evaluation metrics for a sample of high pLDDT, low scPerplexity

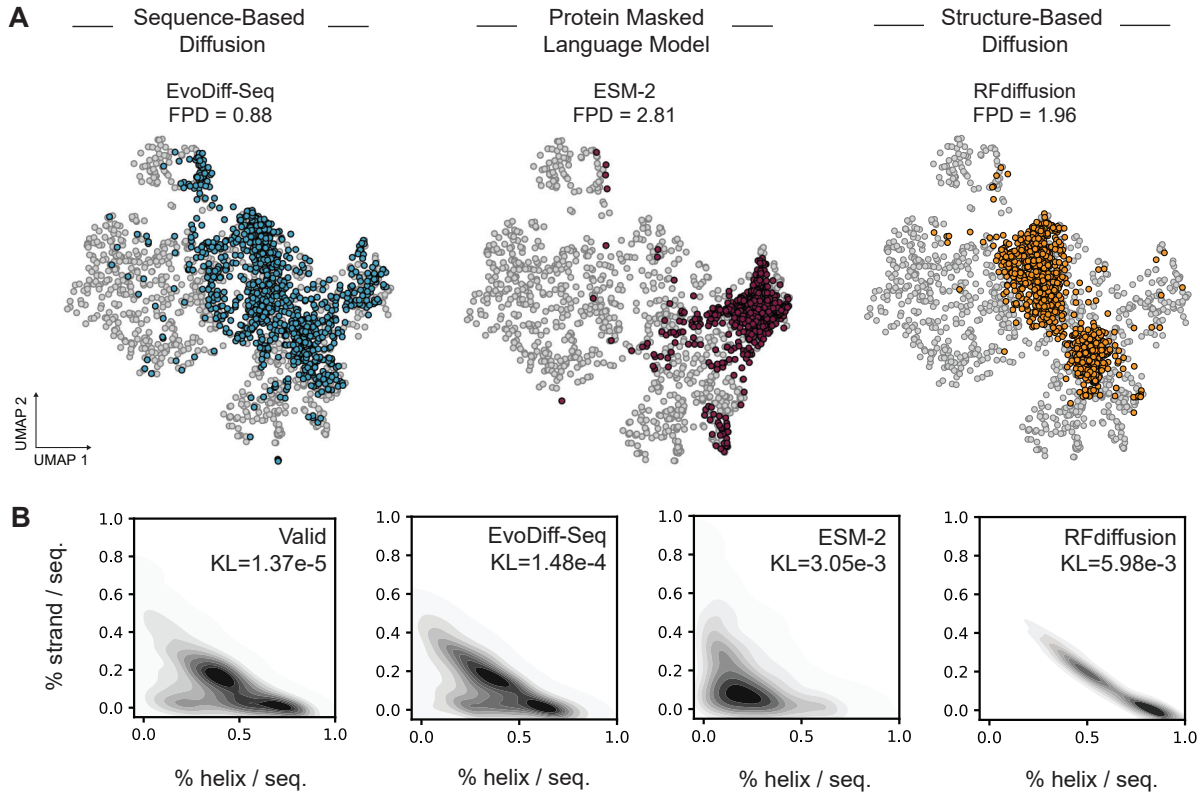

Figure 3: **Generated protein sequences capture natural distributions of protein functional and structural features.** **(A)** UMAP of ProtT5 embeddings, annotated with FPD, of natural sequences from the test set (grey, n=1000) and of generated sequences from EvoDiff-Seq (blue, n=1000) and ESM-2 (red, n=1000), and inferred sequences inverse-folded from structures from RFdiffusion (orange, n=1000). **(B)** Multivariate distributions of helix and strand structural features in generated sequences, based on DSSP 3-state predictions (n=1000 samples from each model or the validation set) and annotated with the Kullback-Leibler (KL) divergence relative to the test set.

conditionally-generated query sequences that exhibit low sequence similarity to anything in the conditioning MSA (**Fig. 4G**). These results indicate that EvoDiff-MSA can conditionally generate novel, structurally plausible members of a protein family given guidance from evolutionary information and without further finetuning.

Generating intrinsically disordered regions Because it generates directly in sequence space, we hypothesized that EvoDiff could natively generate intrinsically disordered regions (IDRs). IDRs are regions within a protein that lack secondary or tertiary structure; up to 30% of eukaryotic proteins contain at least one IDR, and IDRs make up over 40% of the residues in eukaryotic proteomes (*19*). IDRs carry out important and diverse functional roles in the cell directly facilitated by their lack of structure, such as protein-protein interactions (*40, 41*) and signaling (*42*). Altered abundance and mutations in IDRs have been implicated in human disease, including neurodegeneration and cancer (*43–45*). Despite their prevalence and critical roles in function and disease, IDRs do not fit neatly in the structure-function paradigm and remain outside the capabilities of structure-based protein design methods.

Having observed that unconditional generation using EvoDiff-Seq produced a similar fraction of residues predicted to lack secondary structure as that in natural sequences (**Fig. 3B**), we used inpainting with EvoDiff-Seq and EvoDiff-MSA to intentionally generate disordered regions via conditioning on their surrounding structured regions (**Fig. 5A**). To accomplish this, we leveraged a previously curated dataset of computationally predicted IDRs covering the human proteome (*46*). We selected this dataset because it also curates orthologs for these proteins, enabling construction of MSAs (*46*). After using EvoDiff to generate putative IDRs via inpainting, we then predicted disorder scores for each residue in the generated and natural sequences using DR-BERT (*47*) (**Figs. 5A, S10**). Over 100 generations, we observe that IDR regions inpainted by EvoDiff-Seq and EvoDiff-MSA result in distributions of disorder scores similar to those for

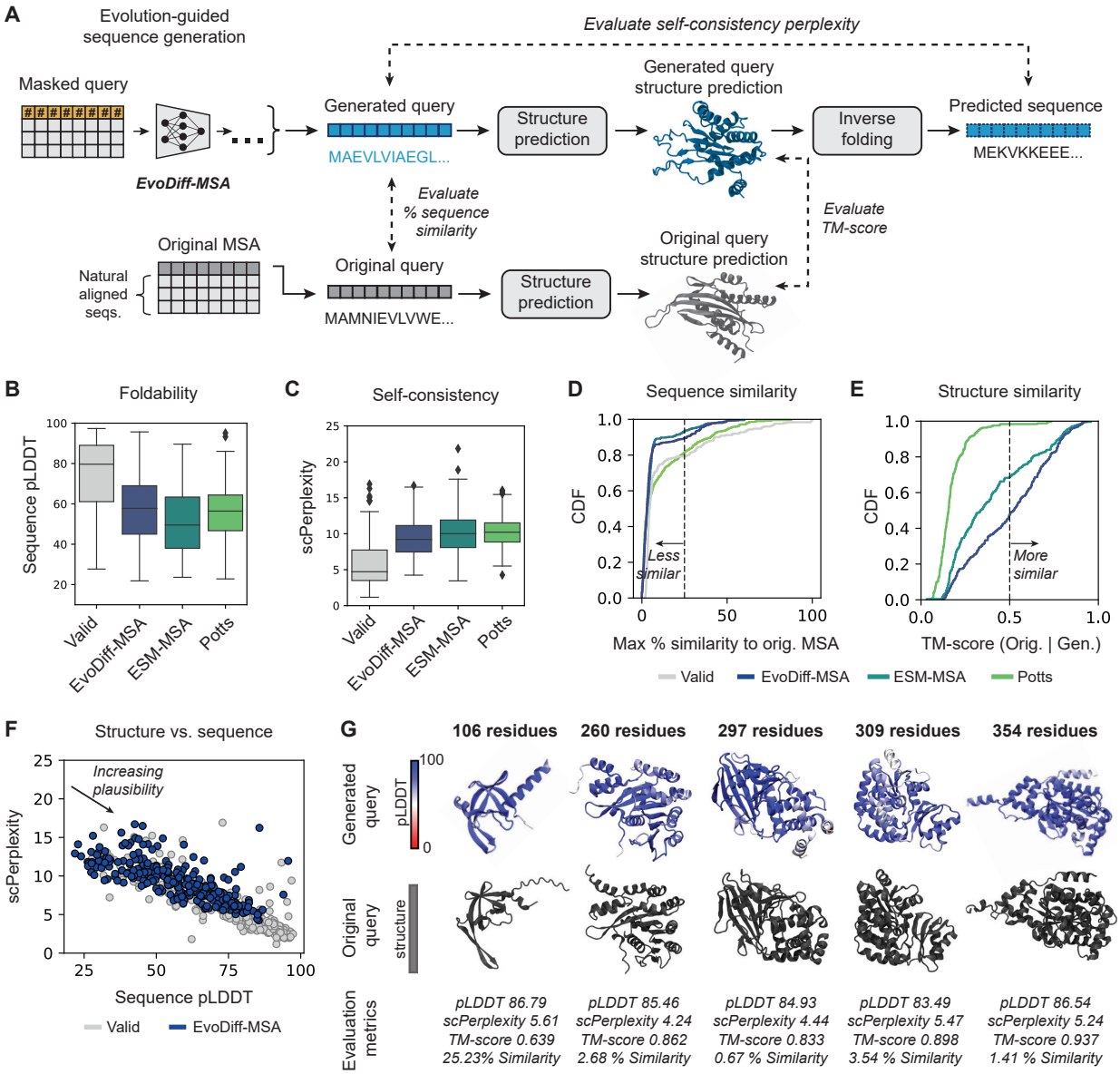

Figure 4: **EvoDiff-MSA enables evolution-guided sequence generation**. **(A)** A new sequence is generated from EvoDiff-MSA via diffusion over only the query component. Generations are evaluated for diversity and self-consistency and for the quality and consistency of their predicted structures. **(B-E)** Distributions of pLDDT (B), scPerplexity (C), sequence similarity (D; dashed line at 25%), and TM-score (E; dashed line at 0.5) for sequences from the validation set, EvoDiff-MSA, ESM-MSA, and a Potts model (n=250 sequences per model; box plots show median and interquartile range). **(F)** Sequence pLDDT versus scPerplexity for sequences from the validation set (grey, n=250) and EvoDiff-MSA (blue, n=250). **(G)** Predicted structures and metrics for structurally plausible generations from EvoDiff-MSA.

natural sequences, across both the IDR and the surrounding structured regions (**Figs. 5B, S11**). Generations from EvoDiff-MSA exhibit strong correlation in predicted disorder scores to those of true IDRs (**Fig. S11**). Although putative IDRs generated by EvoDiff-Seq are less similar to their original IDR than those from EvoDiff-MSA (**Fig. 5C**), both models generated disordered regions that preserve disorder scores over the entire protein sequence and still exhibit low sequence similarity to the original IDR (**Fig. 5D-E**). These results demonstrate that EvoDiff can robustly generate IDRs conditioned on sequence context from surrounding structured regions.

Scaffolding functional motifs with sequence information alone Thus far, the primary application of deep generative models of protein structure in protein engineering is their ability to scaffold binding and catalytic motifs: given the 3D coordinates of a functional motif, these models can often generate a structural scaffold that holds the motif in precisely the 3D geometry needed for function (*10, 14, 48*). Given that the fixed functional motif includes the residue identities for the motif, we investigated whether a structural model is actually necessary for motif scaffolding.

We used conditional generation with EvoDiff to generate scaffolds for a diverse set of 17 motif-scaffolding problems (*10*) by fixing the functional motif, supplying only the motif's amino-acid sequence as conditioning information, and then decoding the remainder of the sequence (**Fig. 1D**). The problems include simple "inpainting", viral epitopes, receptor traps, small molecule binding sites, protein-binding interfaces, and enzyme active sites. Many of the motifs are not contiguous in sequence space. We compared the performance of EvoDiff, which uses only sequence information, to the state-of-the-art structure model RFdiffusion, and facilitated direct comparisons by using OmegaFold to predict structures for our generated sequences as well as for sequences inverse-folded from RFdiffusion structures. Notably, we use the same EvoDiff models for both unconditional and conditional generation, while the version

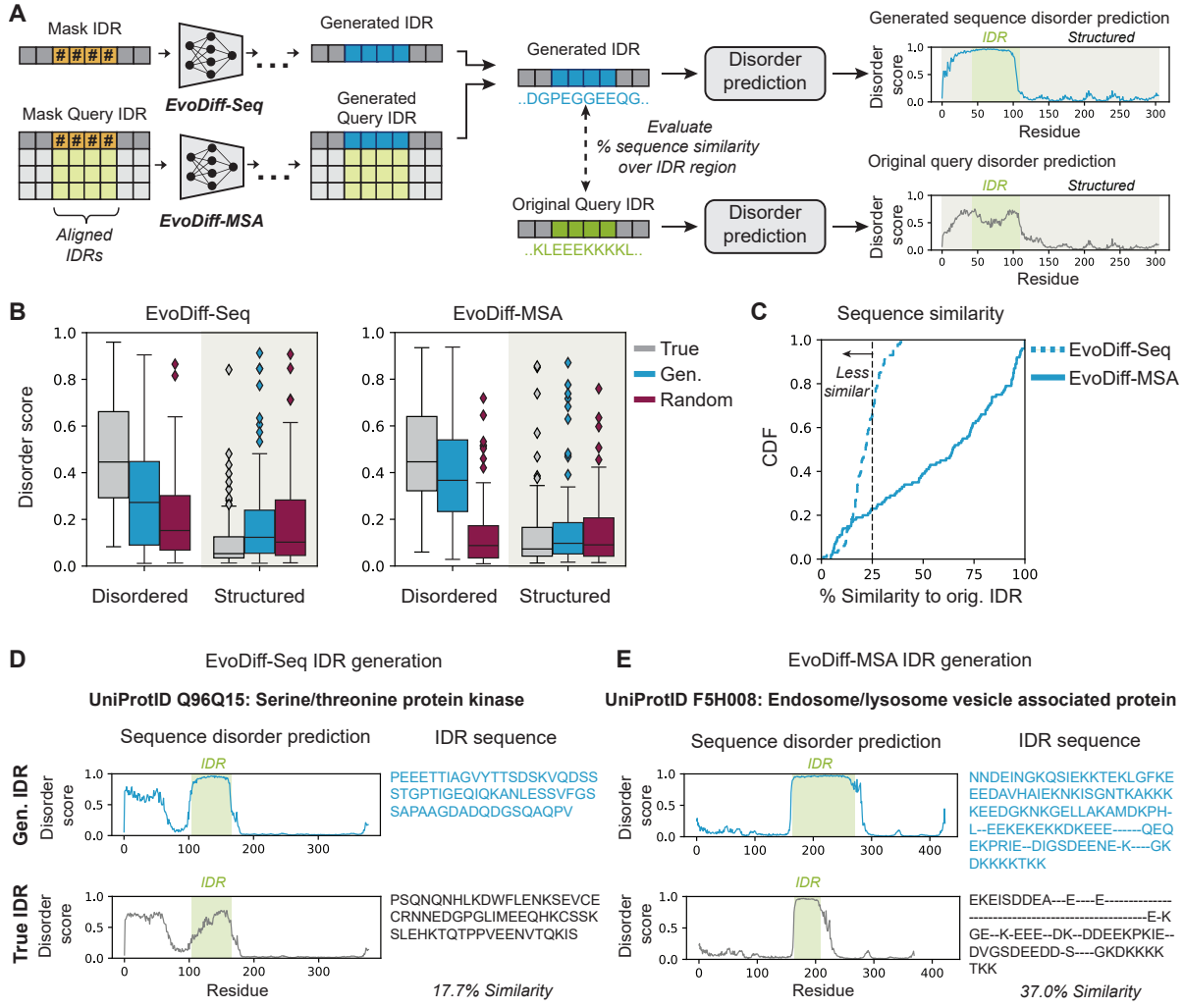

Figure 5: **EvoDiff generates intrinsically disordered regions.** **(A)** A new IDR sequence is generated from EvoDiff-Seq or EvoDiff-MSA by inpainting disordered residues in the query sequence. DR-BERT is then used to predict disorder scores for the original and regenerated sequences. **(B)** Distributions of disorder scores over disordered and structured regions for sequences with true (grey), inpainted (blue), and randomly-sampled (red) IDRs (n=100 sequences per condition; box plots show median and interquartile range). **(C)** Distribution of sequence similarity relative to the original IDR for generated IDRs from EvoDiff-Seq (blue, dashed) and EvoDiff-MSA (blue, solid) (n=100; dashed line at 25%). **(D-E)** Predicted disorder scores and corresponding sequences for representative generated (top row) and true (bottom row) IDRs from EvoDiff-Seq (D) and EvoDiff-MSA (E).

of RFdiffusion used for scaffolding is finetuned from that used for unconditional generation.

We evaluated the ability of each of EvoDiff-Seq, EvoDiff-MSA, and RFdiffusion to generate successful scaffolds (**Fig. 6A-B**), where we define a scaffold to be successful if the predicted motif coordinates have less than 1Å RMSD from the desired motif coordinates. Despite operating entirely in sequence space, EvoDiff-Seq and EvoDiff-MSA generate successful scaffolds for 8 and 13 of the 17 problems, respectively (**Table S5**, **S6**). EvoDiff-MSA has a higher success rate than EvoDiff-Seq for 10 problems and a higher success rate than RFdiffusion for 6 problems. EvoDiff-Seq has a higher success rate than RFdiffusion for 2 problems and a higher success rate than EvoDiff-MSA for 3 problems. There are two scaffolding problems (1YCR, 3IXT) where EvoDiff-MSA is outperformed by both EvoDiff-Seq and RFdiffusion (**Table S5**, **S6**). These are both examples where, for scaffolding, an MSA containing fewer than 64 protein sequences was input to EvoDiff-MSA, which did not see MSAs with fewer than 64 sequences during training.

Interestingly, there is almost no correlation between the problem-specific success rates of EvoDiff and RFdiffusion, and there are very few problems for which both methods have high success rates, showing that EvoDiff may have orthogonal strengths to RFdiffusion (**Fig. 6A-B**). Due to its conditioning on evolutionary information, EvoDiff-MSA generates scaffolds that are more structurally similar to the native scaffold than EvoDiff-Seq (**Fig. 6C**). To ensure that EvoDiff is not finding trivial solutions, we show that it outperforms both random generation and the single-order LRAR model (which decodes unconditionally up to and after a motif) (**Table S5**). ESM-MSA performs similarly to EvoDiff-MSA on this task, as the motif scaffolding task is well-aligned with its training task, and it is trained on approximately 200x more MSAs than EvoDiff-MSA (**Table S6**). We illustrate examples of successful scaffolds sampled from EvoDiff and note both the qualitative and quantitative quality of generated proteins and predicted structures across a range of functional motifs (**Fig. 6D-G**). These results demonstrate

that EvoDiff can design functional scaffolds around structural motifs via conditional generation in sequence space alone.

Discussion

We present EvoDiff, a diffusion modeling framework capable of generating high-fidelity, diverse, and novel proteins with the option of conditioning according to sequence constraints. Because it operates in the universal protein design space, EvoDiff can unconditionally sample diverse structurally-plausible proteins, generate intrinsically disordered regions, and scaffold structural motifs using only sequence information, challenging a paradigm in structure-based protein design.

EvoDiff is the first deep learning framework to demonstrate the power of diffusion generative modeling over evolutionary-scale protein sequence space. Unlike previous attempts to train diffusion models on protein structures (*7–15*) and/or sequences (*49–53*), EvoDiff is trained on a large, diverse sample of all natural sequences, rather than on smaller protein structure datasets or sequence data from a specific protein family. Previous protein generative models trained on global sequence space have been either left-to-right autoregressive (LRAR) models (*54–57*) or masked language models (MLMs) (*27, 28, 30, 37, 58*). EvoDiff's OADM training task generalizes the LRAR and MLM training tasks. Specifically, the OADM setup generalizes LRAR by considering all possible decoding orders, while the MLM training task is equivalent to training on one step of the OADM diffusion process.

This generalized mathematical formulation yields empirical benefits, as EvoDiff-Seq produces sequences that better cover protein functional and structural space than sampling from state-of-the-art protein MLMs (**Fig. 3**). While an LRAR model learned to fit the evolutionary sequence distribution better (**Table S1**), the fixed decoding order of traditional left-to-right autoregression cannot be used to perform conditional generation. EvoDiff directly addresses

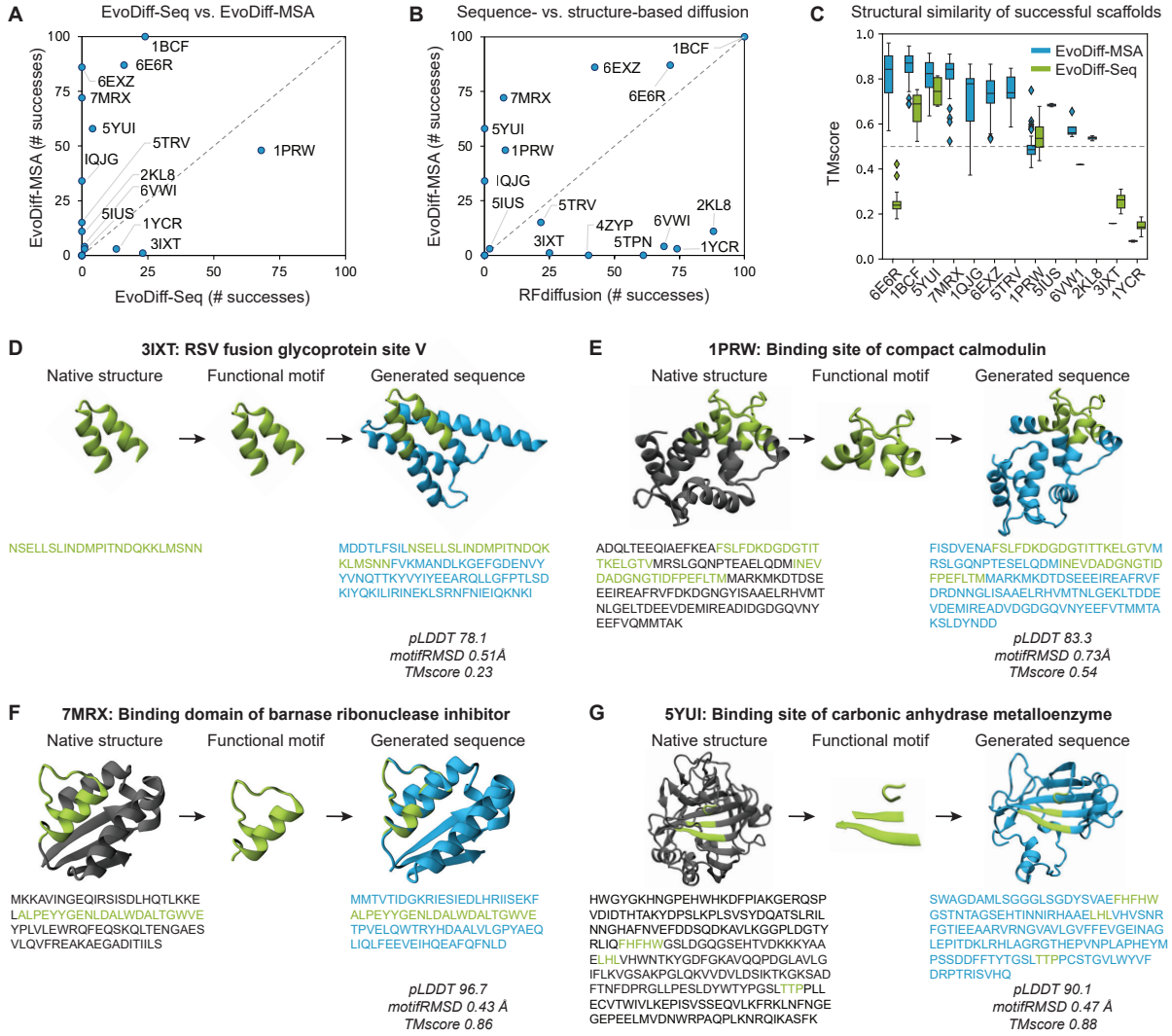

Figure 6: **EvoDiff scaffolds functional motifs without explicit structural information**. **(A)** Number and identity of successful scaffolds from n=100 trials for EvoDiff-Seq (x-axis) versus EvoDiff-MSA (y-axis) across scaffolding problems in which at least one method succeeds. **(B)** Performance comparison of sequence-based scaffolding via EvoDiff-MSA (y-axis) versus structure-based scaffolding via RFdiffusion (x-axis) across scaffolding problems in which at least one method succeeds (n=100 trials per problem). **(C)** Distributions of TM-scores of successfully generated scaffolds from EvoDiff models relative to the true structures (dashed line at 0.5; box plots show median and interquartile range). **(D-G)** Generated sequences, predicted structures, and computed metrics for representative scaffolding examples from EvoDiff-Seq (D-E) and EvoDiff-MSA (F-G). Motif is shown in green, original scaffold in black, and generated scaffold in blue.

this barrier by enabling different forms of conditioning, including evolution-guided generation (**Fig. 4**) as well as inpainting and scaffolding (**Figs. 5-6**). We report the first demonstrations of these programmable generation capabilities from deep generative models of protein sequence alone.

Future work may expand these capabilities to enable conditioning via guidance, in which generated sequences can be iteratively refined to fit desired properties. While we observe that OADM generally outperforms D3PM in unconditional generation, likely because the OADM denoising task is easier to learn than that of D3PM, conditioning via guidance intuitively fits into the EvoDiff-D3PM framework because the identity of each residue in a sequence can be edited at every decoding step. OADM and existing conditional LRAR models, such as ProGen (*54*), both fix the identity of each amino acid once it is decoded, limiting the effectiveness of guidance. Guidance-based conditioning of EvoDiff-D3PM should enable the generation of new protein sequences specifying functional objectives, such as those specified by sequence-function classifiers.

Because EvoDiff only requires sequence data, it can readily be extended for diverse downstream applications, including those not reachable from a traditional structure-based paradigm. As a first example, we have demonstrated EvoDiff's ability to generate IDRs – overcoming a prototypical failure mode of structure-based predictive and generative models – via inpainting without fine-tuning. Fine-tuning EvoDiff on application-specific datasets, such as those from display libraries or large-scale screens, may unlock new biological, therapeutic, or scientific design opportunities that would be otherwise inaccessible due to the cost of obtaining structures for large sequence datasets. Experimental data for structures is much sparser compared to sequences, and while structures for many sequences can be predicted using AlphaFold and similar algorithms, these methods do not work well on point mutants and can be overconfident on spurious proteins (*59, 60*).

While we demonstrated some coarse-grained strategies for conditioning generation through scaffolding and inpainting, to achieve even more fine-grained control over protein function, with future development EvoDiff may be conditioned on text, chemical information, or other modalities. For example, text-based conditioning (*61*) could be used to ensure that generated proteins are soluble, readily expressed, and non-immunogenic. Future use cases for this vision of controllable protein sequence design include programmable modulation of nucleic acids via conditionally-designed transcription factors or endonucleases, improved therapeutic windows via biologics optimized for *in vivo* delivery and trafficking, as well as newly-enabled catalysis via zero-shot tuning of enzyme substrate specificity.

In summary, we present an open-source suite of discrete diffusion models that provide a foundation for sequence-based protein engineering and design. EvoDiff models can be directly deployed for unconditional, evolution-guided, and conditional generation of protein sequences and may be extended for guided design based on structure or function. We envision that EvoDiff will enable new abilities in controllable protein design by reading and writing function directly in the language of proteins.

References

1. Z. Wu, K. E. Johnston, F. H. Arnold, K. K. Yang, *Current Opinion in Chemical Biology* **65**, 18–27 (2021). Protein sequence design with deep generative models.

2. S. L. Lovelock, *et al.*, *Nature* **606**, 49–58 (2022). The road to fully programmable protein catalysis.

3. J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, *International Conference on Machine Learning* (PMLR, 2015), pp. 2256–2265. Deep unsupervised learning using nonequilibrium thermodynamics.

4. P. Dhariwal, A. Nichol, *Advances in Neural Information Processing Systems* **34**, 8780–8794 (2021). Diffusion models beat GANs on image synthesis.

5. J. Ho, A. Jain, P. Abbeel, *Advances in Neural Information Processing Systems* **33**, 6840-6851 (2020). Denoising diffusion probabilistic models.

6. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 10684–10695 (2022). High-resolution image synthesis with latent diffusion models.

7. N. Anand, T. Achim, *arXiv* **2205.15019** (2022). Protein structure and sequence generation with equivariant denoising diffusion probabilistic models.

8. K. E. Wu, *et al.*, *arXiv* **2209.15611** (2022). Protein structure generation via folding diffusion.

9. B. L. Trippe, *et al.*, *The Eleventh International Conference on Learning Representations* **11** (2023). Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem.

10. J. L. Watson, *et al.*, *Nature* **620**, 1089-1100 (2023). De novo design of protein structure and function with RFdiffusion.

11. J. Ingraham, *et al.*, *bioRxiv* **2022.12.01.518682** (2022). Illuminating protein space with a programmable generative model.

12. Y. Lin, M. AlQuraishi, *Proceedings of the 40th International Conference on Machine Learning* (2023). Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds.

13. J. Yim, *et al.*, *arXiv preprint arXiv:2302.02277* (2023). SE (3) diffusion model with application to protein backbone generation.

14. J. S. Lee, J. Kim, P. M. Kim, *Nature Computational Science* **3**, 382–392 (2023). Score-based generative modeling for de novo protein design.

15. A. E. Chu, L. Cheng, G. El Nesr, M. Xu, P.-S. Huang, *bioRxiv* (2023). An all-atom protein generative model.

16. N. Tokuriki, D. S. Tawfik, *Science* **324**, 203–207 (2009). Protein dynamism and evolvability.

17. P. A. Romero, F. H. Arnold, *Nature Reviews Molecular Cell Biology* **10**, 866–876 (2009). Exploring protein fitness landscapes by directed evolution.

18. J. Gao, D. G. Truhlar, *Annual Review of Physical Chemistry* **53**, 467–505 (2002). Quantum mechanical methods for enzyme kinetics.

19. N. E. Davey, *Current Opinion in Structural Biology* **56**, 155–163 (2019). The functional importance of structure in unstructured protein regions.

20. H. Frauenfelder, S. G. Sligar, P. G. Wolynes, *Science* **254**, 1598–1603 (1991). The energy landscapes and motions of proteins.

21. J. McCammon, *Reports on Progress in Physics* **47**, 1 (1984). Protein dynamics.

22. K. Henzler-Wildman, D. Kern, *Nature* **450**, 964–972 (2007). Dynamic personalities of proteins.

23. P. K. Agarwal, *Journal of the American Chemical Society* **127**, 15248–15256 (2005). Role of protein dynamics in reaction rate enhancement by enzymes.

24. E. Hoogeboom, *et al.*, *The Eleventh International Conference on Learning Representations* **11** (2022). Autoregressive diffusion models.

25. J. Austin, D. D. Johnson, J. Ho, D. Tarlow, R. van den Berg, *Advances in Neural Information Processing Systems* **34** (2021). Structured denoising diffusion models in discrete state-spaces.

26. B. E. Suzek, *et al.*, *Bioinformatics* **31**, 926–932 (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches.

27. K. K. Yang, N. Fusi, A. X. Lu, *bioRxiv* (2022). Convolutions are competitive with transformers for protein sequence pretraining.

28. R. M. Rao, *et al.*, *Proceedings of the 38th International Conference on Machine Learning* **139**, 8844–8856 (2021). MSA Transformer.

29. G. Ahdritz, *et al.*, *bioRxiv* (2022). OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization.

30. R. Verkuil, *et al.*, *bioRxiv* (2022). Language models generalize beyond natural proteins.

31. R. Wu, *et al.*, *bioRxiv* (2022). High-resolution de novo structure prediction from primary sequence.

32. K. M. Ruff, R. V. Pappu, *Journal of Molecular Biology* **433**, 167208 (2021). AlphaFold and implications for intrinsically disordered proteins.

33. C. Hsu, *et al.*, *Proceedings of the 39th International Conference on Machine Learning* **162**, 8946–8970 (2022). Learning inverse folding from millions of predicted structures.

34. J. Dauparas, *et al.*, *Science* **378**, 49-56 (2022). Robust deep learning–based protein sequence design using ProteinMPNN.

35. M. Littmann, M. Heinzinger, C. Dallago, T. Olenyi, B. Rost, *Scientific Reports* **11**, 1160 (2021). Embeddings from deep learning transfer GO annotations beyond homology.

36. Gene Ontology Consortium, *Nucleic Acids Research* **47**, D330–D338 (2019). The gene ontology resource: 20 years and still GOing strong.

37. A. Elnaggar, *et al.*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 7112–7127 (2021). ProtTrans: Toward understanding the language of life through self-supervised learning.

38. W. Kabsch, C. Sander, *Biopolymers: Original Research on Biomolecules* **22**, 2577–2637 (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.

39. S. Vorberg, S. Seemayer, J. Söding, *PLOS Computational Biology* **14**, e1006526 (2018). Synthetic protein alignments by CCMgen quantify noise in residue-residue contact prediction.

40. V. N. Uversky, *Advances in Protein Chemistry and Structural Biology* **110**, 85–121 (2018). Intrinsic disorder, protein–protein interactions, and disease.

41. B. Mészáros, I. Simon, Z. Dosztányi, *Physical Biology* **8**, 035003 (2011). The expanding view of protein–protein interactions: complexes involving intrinsically disordered proteins.

42. P. E. Wright, H. J. Dyson, *Nature Reviews Molecular Cell Biology* **16**, 18–29 (2015). Intrinsically disordered proteins in cellular signalling and regulation.

43. V. Vacic, *et al.*, *PLOS Computational Biology* **8**, 1-14 (2012). Disease-associated mutations disrupt functionally important regions of intrinsic protein disorder.

44. O. Coskuner-Weber, O. Mirzanli, V. N. Uversky, *Biophysical Reviews* **14**, 679–707 (2022). Intrinsically disordered proteins and proteins with intrinsically disordered regions in neurodegenerative diseases.

45. L. M. Iakoucheva, C. J. Brown, J. D. Lawson, Z. Obradović, A. K. Dunker, *Journal of Molecular Biology* **323**, 573–584 (2002). Intrinsic disorder in cell-signaling and cancer-associated proteins.

46. A. X. Lu, *et al.*, *PLOS Computational Biology* **18**, e1010238 (2022). Discovering molecular features of intrinsically disordered regions by using evolution for contrastive learning.

47. A. Nambiar, J. M. Forsyth, S. Liu, S. Maslov, *bioRxiv* (2023). DR-BERT: A protein language model to annotate disordered regions.

48. J. Wang, *et al.*, *Science* **377**, 387–394 (2022). Scaffolding protein functional sites using deep learning.

49. Z. Jiang, *et al.*, *bioRxiv* (2023). PRO-LDM: Protein sequence generation with conditional latent diffusion models.

50. B. Zhou, *et al.*, *bioRxiv* (2023). Conditional protein denoising diffusion generates programmable endonucleases.

51. N. Gruver, *et al.*, *arXiv* **2305.20009** (2023). Protein design with guided discrete diffusion.

52. S. L. Lisanza, *et al.*, *bioRxiv* (2023). Joint generation of protein sequence and structure with RoseTTAFold sequence space diffusion.

53. C. Shi, C. Wang, J. Lu, B. Zhong, J. Tang, *The Eleventh International Conference on Learning Representations* (2022). Protein sequence and structure co-design with equivariant translation.

54. A. Madani, *et al.*, *Nature Biotechnology* **41**, 1099–1106 (2023). Large language models generate functional protein sequences across diverse families.

55. N. Ferruz, S. Schmidt, B. Höcker, *Nature Communications* **13**, 4348 (2022). ProtGPT2 is a deep unsupervised language model for protein design.

56. T. F. Truong Jr, T. Bepler, *arXiv preprint arXiv:2306.06156* (2023). PoET: A generative model of protein families as sequences-of-sequences.

57. L. Zhang, J. Chen, T. Shen, Y. Li, S. Sun, *arXiv preprint arXiv:2306.01824* (2023). Enhancing the protein tertiary structure prediction by multiple sequence alignment generation.

58. A. Rives, *et al.*, *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences.

59. V. Monzon, D. H. Haft, A. Bateman, *Bioinformatics Advances* **2**, vbab043 (2022). Folding the unfoldable: using AlphaFold to explore spurious proteins.

60. M. A. Pak, *et al.*, *PLOS One* **18**, e0282689 (2023). Using AlphaFold to predict the impact of single mutations on protein stability and function.

61. S. Liu, *et al.*, *arXiv* **2302.04611** (2023). A text-guided protein design framework.

62. E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, M. Welling, *arXiv* **2102.05379** (2021). Argmax flows and multinomial diffusion: Learning categorical distributions.

63. J. Song, C. Meng, S. Ermon, *arXiv* **2010.02502** (2020). Denoising diffusion implicit models.

64. S. Henikoff, J. G. Henikoff, *Proceedings of the National Academy of Sciences* **89**, 10915–10919 (1992). Amino acid substitution matrices from protein blocks.

65. N. Kalchbrenner, *et al.*, *arXiv* **1610.10099** (2017). Neural machine translation in linear time.

66. A. Paszke, *et al.*, *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019), pp. 8024–8035.

67. A. Vaswani, *et al.*, *arXiv* **1706.03762** (2017). Attention is all you need.

68. D. P. Kingma, J. Ba, *arXiv* **1412.6980** (2017). Adam: A method for stochastic optimization.

69. N. Ferruz, *et al.*, *Computational and Structural Biotechnology Journal* **21**, 238-250 (2023). From sequence to function through structure: deep learning for protein design.

70. Y. Zhang, J. Skolnick, *Proteins: Structure, Function, and Bioinformatics* **57**, 702–710 (2004). Scoring function for automated assessment of protein structure template quality.

71. J. Hanson, Y. Yang, K. Paliwal, Y. Zhou, *Bioinformatics* **33**, 685–692 (2017). Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks.

72. A. M. Altenhoff, *et al.*, *Nucleic Acids Research* **49**, D373–D379 (2021). OMA orthology in 2021: website overhaul, conserved isoforms, ancestral gene order and more.

73. J. Jumper, *et al.*, *Nature* **596**, 583-589 (2021). Highly accurate protein structure prediction with AlphaFold.

Acknowledgements

The authors thank Christian Dallago for helpful discussions about the methods, applications, and evaluations; Jonathan Carlson for valuable feedback on the manuscript; Kevin E. Wu for providing sequences generated from FoldingDiff; Joseph Watson and David Juergens for providing sequences generated from RFDiffusion; and Alessandro Sordoni, Hannes Schultz, Rémi Piché-Taillefer, and Remi Tachet des Combes for assistance on using Microsoft's compute resources.

Author contributions

Conceptualization: S.A., N.T., N.F., A.P.A., K.K.Y.; Methodology: S.A., N.T., R.vdB., A.P.A., K.K.Y.; Software Programming: S.A., N.T., K.K.Y.; Validation: S.A., N.T., A.X.L., A.P.A., K.K.Y.; Formal analysis: S.A., N.T., A.X.L., K.K.Y.; Resources Provision: N.F., K.K.Y.; Data Curation: S.A., N.T., A.X.L., K.K.Y.; Visualization: S.A., A.P.A., K.K.Y; Writing - Original Draft: S.A., A.P.A., K.K.Y.; Writing - Review & Editing: S.A., N.T., R.vdB., A.X.L., N.F., A.P.A., K.K.Y.; Supervision: N.F., A.P.A., K.K.Y.

Resource availability

Code is available at `https://github.com/microsoft/evodiff`. Model weights, generated sequences, and computed metrics are available at `https://zenodo.org/record/8332830`.

Methods

Diffusion models Diffusion models are a class of generative models that learn to generate data from noise. They consist of a forward corruption process and a learned reverse denoising process. The forward process is a Markov chain of diffusion steps $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that corrupts an input ($\boldsymbol{x}_0$) over T timesteps such that $\boldsymbol{x}_T$ is indistinguishable from random noise. The learned reverse denoising process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is parameterized by a model such as a neural network and generates new data from noise. Discrete diffusion models have previously been developed over binary random variables (*3*), developed over categorical random variables with uniform transition matrices (*62, 63*), linked to autoregressive models (*24*), and optimized for use with transition matrices (*25*).

This work presents models from two different discrete diffusion frameworks – order-agnostic autoregressive diffusion models (OADMs) and discrete denoising diffusion probabilistic models (D3PMs) – on protein sequences and multiple sequence alignments (MSAs).

Discrete Denoising Diffusion Probabilistic Models (D3PMs) Discrete denoising diffusion probabilistic models (D3PMs) operate by defining a transition matrix Q such that, over T timesteps, discrete inputs (i.e. protein amino-acid sequences for EvoDiff) are iteratively corrupted via a controlled Markov process until they constitute samples from a uniform stationary distribution at time T. This section describes the D3PM process and loss for a single categorical variable x in one-hot format. The forward corruption process is described by:

$$q(x_t|x_{t-1}) = \mathrm{Cat}(x_t; p = x_{t-1}Q_t). \tag{1}$$

This allows for efficient training via efficient computation of $q(x_t|x_0)$ and $q(x_{t-1}|x_t)$. The D3PM approach can emulate a masked modeling process by choosing a transition matrix with an absorbing state (e.g., [MASK]; (*25*)). However, in this work, the D3PM formulation is only

used for discrete corruption because masking corruption via OADM generally outperforms absorbing-state D3PM (*24*). EvoDiff includes two discrete corruption schemes: one based on a uniform transition matrix (D3PM-Uniform) and one based on a biologically-informed transition matrix (D3PM-BLOSUM).

EvoDiff-D3PM models are trained via a hybrid loss function

$$\mathcal{L}_\lambda = \mathcal{L}_{vb} + \lambda \mathcal{L}_{ce}. \tag{2}$$

This loss combines a variational lower bound $\mathcal{L}_{vb}$ on the negative log likelihood

$$\mathcal{L}_{vb} = \mathbb{E}_{q(x_0)}[\underbrace{D_{KL}\left[q(x_T|x_0)\|p(x_T)\right]}_{\mathcal{L}_T} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(x_t|x_0)}\left[D_{KL}\left[q(x_{t-1}|x_t,x_0)\|p_\theta(x_{t-1}|x_t)\right]\right]}_{\mathcal{L}_{t-1}} + \underbrace{-\mathbb{E}_{q(x_1|x_0)}\left[\log(p_\theta(x_0|x_1)\right]]}_{\mathcal{L}_0}. \tag{3}$$

and a cross-entropy loss $\mathcal{L}_{ce}$ on $p_\theta(x_0|x_t)$. Investigation of the impact of λ on model performance revealed minimal improvement to sample generation quality when $\lambda > 0$, consistent with the findings of the original D3PM paper (*25*). Thus λ=0 and T=500 were used in all D3PM experiments.

$\mathcal{L}_{vb}$ has three terms. $\mathcal{L}_T$ measures whether the corruption reaches the stationary distribution $p(x_T)$ at time T and does not depend on θ. Next consider the remaining two terms $\mathcal{L}_{t-1}$ and $\mathcal{L}_0$, which depend on θ. Following the original D3PM paper, $\tilde{p}_\theta(\tilde{x}_0|x_t)$ is directly predicted by the neural network. To compute the loss at timesteps $0 < t < T$, the terms $q(x_{t-1}|x_t, x_0)$ and $p_\theta(x_{t-1}|x_t)$ must be computed from x_t, x_0, and $\tilde{p}_\theta(\tilde{x}_0|x_t)$ using Markov properties

Defining $\overline{Q}_t = Q_1 Q_2 \cdots Q_t$:

$$q(x_{t-1}|x_t, x_0) = \text{Cat}\left(x_{t-1}; p = \frac{x_t Q_t^\top \odot x_0 \overline{Q}_{t-1}}{x_0 \overline{Q}_t x_t^\top}\right) \tag{4}$$

$$p_\theta(x_{t-1}|x_t) \propto \sum_{\tilde{x}_0} q(x_{t-1}, x_t|\tilde{x}_0)\tilde{p}_\theta(\tilde{x}_0|x_t) \tag{5}$$

where $\odot$ represents an element-wise product. For Equation 5 rules of conditional probability and Markov properties are used to define $q(x_{t-1}, x_t|\tilde{x}_0)$ in terms of x_t and $\tilde{x}_0$:

$$q(x_{t-1}, x_t|\tilde{x}_0) = \text{Cat}\left(x_{t-1}; p = x_t Q_t^\top \odot \tilde{x}_0 \overline{Q}_{t-1}\right) \tag{6}$$

Putting everything together, at each step of training a corruption timestep is sampled according to $t \sim \mathcal{U}(1, \ldots, T-1)$. x_t is then sampled via $q(x_t|x_0) \sim \text{Cat}\left(x_t; p = x_0 \overline{Q}_t\right)$ for every residue in the input protein, and the neural network predicts $\tilde{p}_\theta(\tilde{x}_0|x_t)$. Note that, while the corruption and loss are computed independently over each residue, the neural network predicts $\tilde{p}$ in the context of the entire sequence. If $t = 1$, only the loss $\mathcal{L}_0$ is used, reflecting a standard negative log likelihood. Otherwise, Equations 4 and 5 are used to compute the loss $\mathcal{L}_{t-1}$.

Sampling from a trained model begins with the noised $\boldsymbol{x}_T$, where each residue is randomly sampled from a uniform distribution over amino acids. x_{t-1} is then iteratively sampled via $p_\theta(x_{t-1}|x_t)$ as described in Equation 5. For all models, generated sequences are sampled to match the distribution of sequence lengths in the training set, going up to 2048 residues as the maximum length.

EvoDiff-D3PM-Uniform Many strategies exists to schedule corruption in D3PMs. EvoDiff-D3PM-Uniform employs the simplest case – a uniform corruption scheme. Specifically, EvoDiff-D3PM-Uniform models implement a doubly stochastic, uniform transition matrix Q_t with a corruption schedule $(T - t + 1)^{-1}$ from Sohl-Dickstein et al. (*3*), so that information is linearly corrupted between x_t and x_0 for all $t < T$.

EvoDiff-D3PM-BLOSUM EvoDiff-D3PM-BLOSUM implements a transition matrix derived from BLOSUM62 matrices of amino acid substitution frequencies (*64*). BLOSUM matrices are derived from observed alignments across highly conserved regions of protein families and thus

provide the relative frequencies of amino acids and their substitution probabilities.

Rows that represent uniform transition probabilities for non-standard amino acid codes (J, O, U) and for $<$ GAP $>$ tokens in the MSA input case are included in addition to standard amino acids. BLOSUM substitution frequencies are converted to a matrix of transition probabilities by performing a Softmax over the frequencies and then normalizing over rows and columns via the Sinkhorn-Knopp algorithm to obtain a doubly stochastic matrix. In this scheme, the gradual corruption of a single sequence to random noise is simulated in a way that prioritizes conserved evolutionary relationships of amino acid mutations. A β-schedule was implemented to taper the number of mutations over time for timesteps up to T=500, specifically via an empirical schedule that corrupts half the sequence content by half of T (t=250) (**Fig. S12**). This schedule was chosen to approximate the linear rate of mutations observed over 500 timesteps in the uniform transition matrix case, shown in Fig. S12b.

Order-Agnostic Autoregressive Diffusion Models (OADMs) Order-agnostic autoregressive diffusion models (OADMs) generalize absorbing-state D3PM and left-to-right autoregressive models (LRARs) (*24*). This section describes the OADM process and loss for a sequence $\boldsymbol{x}$ of L categorical variables. In the case of EvoDiff, L is the sequence length.

LRARs factorize a high-dimensional joint distribution $p(\boldsymbol{x})$ into the product of L univariate distributions using the probability chain rule:

$$\log p(\boldsymbol{x}) = \sum_{t=1}^{L} \log p(x_t|\boldsymbol{x}_{<t}) \tag{7}$$

where $\boldsymbol{x}_{<t} = x_1, x_2, \ldots x_{t-1}$. LRARs are typically parametrized using a triangular dependency structure, such as causal masking in a transformer or CNN, in order to allow parallelized computation of all the conditional distributions in the likelihood during training. LRARs learn to generate sequences in a pre-specified left-to-right decoding order, which may be non-obvious for modalities such as proteins and does not allow conditioning on arbitrary fixed subsequences.

LRARs can be expanded into a diffusion framework via two subtle changes. Following the exposition in Hoogeboom *et al.,* (*24*), the first change is to allow order-agnostic decoding. In an order-agnostic autoregressive model, a decoding order σ is first sampled uniformly from all possible decoding orders S_L. At time step t in the forward process, $x_{\sigma(L-t)}$ is masked. The log-likelihood for an order-agnostic autoregressive model is derived using Jensen's inequality:

$$\begin{aligned}\log p(\boldsymbol{x}) = \log \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} p(\boldsymbol{x}|\sigma) &\geq \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} \log p(\boldsymbol{x}|\sigma) \\ &\geq \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} \sum_{t=1}^{L} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)})\end{aligned}$$

The next change involves an objective that optimizes over arbitrary decoding orders one timestep at a time in the style of modern diffusion models, without requiring a neural network that enforces a triangular or causal dependency structure. This is accomplished by replacing the summation over t by an expectation that is appropriately re-weighted.

$$\begin{aligned}\log p(\boldsymbol{x}) &\geq \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} \sum_{p=1}^{L} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)}) \\ &= \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} L \cdot \mathbb{E}_{t\sim\mathcal{U}(1,\ldots,L)} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)}) \\ &= L \cdot \mathbb{E}_{t\sim\mathcal{U}(1,\ldots,L)} \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} \frac{1}{L-t+1} \sum_{k\in\sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)})\end{aligned}$$

The overall expected log likelihood $\log p(\boldsymbol{x})$ can be thought of according to a series of likelihoods, each captured in the loss at step t, $\mathcal{L}_t$:

$$\mathcal{L}_t = \frac{1}{L-t+1} \mathbb{E}_{\sigma\sim\mathcal{U}(S_L)} \sum_{k\in\sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)}). \tag{8}$$

Thus, the overall expected log likelihood is lower bounded as:

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{t\sim\mathcal{U}(1,\ldots,L)} \left[L \cdot \mathcal{L}_t\right] \tag{9}$$

A neural network can be efficiently trained to learn the reverse process $p_\theta(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)})$ by randomly masking a set of t tokens at each iteration and minimizing the reweighted loss, allowing the model to learn from predictions of all masked positions at each timestep. By learning

one model over all possible decoding orders, OADM allows for conditioning by fixing arbitrary subsequences at generation time. Sequences were generated unconditionally from OADM models by beginning with an all-mask sequence as input, randomly sampling a decoding order, and sampling each token from the predicted probability distribution.

Left-to-right autoregressive and masked language models are diffusion models The connection between autoregressive models and diffusion models has been described previously (*24, 25*). Left-to-right autoregressive (LRAR) diffusion models implement a masked modeling process that is akin to a process which iteratively and deterministically masks all tokens to the right of the sampled token x_t, where the current diffusion timestep t is equivalent to the number of tokens masked over the entire sequence length, with all tokens masked at the final timestep $T = L$.

Likewise, masked language models (MLMs) are equivalent to only learning one step t of OADM:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{L-t+1}\mathbb{E}_{\sigma\sim\mathcal{U}(S_L)}\sum_{k\in\sigma(\geq t)}\log p(x_k|\boldsymbol{x}_{\sigma(<t)}). \tag{10}$$

Thus, the OADM setup generalizes LRAR models by considering all possible decoding orders rather than left-to-right decoding, while the MLM learning task is equivalent to only training on one step of the OADM diffusion process.

Datasets Sequence-only EvoDiff models were trained on UniRef50 (*26*) which contains approximately 45 million protein sequences. The UniRef50 release and train/validation/testing splits from CARP (*27*) were used to facilitate comparisons between models. Sequences longer than 1024 residues were randomly subsampled to 1024 residues. Multiple sequence alignment (MSA) EvoDiff models were trained on OpenFold (*29*), which contains 401,381 MSAs for 140,000 unique Protein Data Bank (PDB) chains and 16,000,000 UniClust30 clusters. To

construct the MSAs used to train EvoDiff, lowercase characters were removed to restore the alignments, as the queries do not contain gap characters. Next, MSAs that contained sequences with more than 512 consecutive $<$ GAP $>$ tokens as well as MSAs that contained fewer than 64 sequences per alignment were filtered out. This filtering resulted in 382,296 total MSAs, which were then randomly split into 372,296 training and 10,000 validation MSAs.

MSA subsampling for training EvoDiff-MSA models To optimize for memory constraints during training, MSAs were subsampled to 64 sequences and a maximum sequence length of 512. MSAs shorter than 512 sequences were padded to a sequence length of 512, but MSAs containing fewer than 64 sequences were excluded from training. For MSAs with more than 64 sequences, two subsampling schemes were implemented: random ("Rand.") and MaxHamming ("Max"). The random subsampling scheme ("Rand.") randomly samples 64 sequences from the MSA, making sure that the reference/query sequence (i.e. the first sequence) is always included. The "Max" subsampling scheme greedily selects for sequence diversity in the 64 sequence subset by iteratively selecting the sequence that maximizes the minimum Hamming distance to the sequences already selected. The Hamming distance measures the distance between two sequences, denoted by the number of amino acids that differ between aligned sequences. Subsampling to maximize the Hamming distance enabled input of an MSA rich with evolutionarily diverse sequences to EvoDiff-MSA models.

Modeling, architecture, and training details For sequences, the EvoDiff denoising model adopts a ByteNet-style CNN architecture (*65*) previously shown to perform similarly to transformers for protein sequence masked language modeling tasks (*27*). All models are implemented in PyTorch (*66*). In EvoDiff-OADM models, the diffusion timestep is implicitly encoded in the number of masked positions. EvoDiff-D3PM models use a 1D sinusoidal encoding (*67*) to denote the timestep for each input. All sequence models were trained with the Adam

optimizer (*68*), a learning rate of 1e-4 with linear warmup over 16,000 steps, and dynamic batching to optimize GPU usage. EvoDiff's small sequence models implement a ByteNet-style architecture with ca. 38M parameters. Large models were scaled to a ByteNet architecture of ca. 640M parameters by increasing the model dimension d from 1020 to 1280, increasing the encoder hidden dimension from $d/2$ to d, and increasing the number of layers from 16 to 56.

38M parameter models were trained on 8 32GB NVIDIA V100 GPUs; 640M parameter models were trained on 32 (2x16) 32GB NVIDIA V100 GPUs. The maximum number of tokens per GPU in each batch was reduced from 40,000 to 6,000 to accommodate training the larger 640M parameter models. 38M parameter models were trained for approximately 2 weeks and saw ca. 3e14 tokens over 700,000 training steps. 640M parameter models were trained for as long as computationally feasible to achieve the best results possible; models saw between ca. 1e10 and 1e17 tokens over ca. 400,000-2,000,000 training steps. The D3PM-BLOSUM model stopped improving after approximately 12 days of training. The D3PM-Uniform and OADM models were trained for 23 days without reaching convergence.

For MSAs, the EvoDiff denoising model adopts a 100M parameter MSA Transformer architecture (*28*). As with the single sequence models, EvoDiff-MSA-OADM models implicitly encode the diffusion timestep; EvoDiff-MSA-D3PM models include an additional sinusoidal timestep embedding. All MSA models were trained with the Adam optimizer with a learning rate of 1e-4 and linear warmup over 15,000 steps. EvoDiff MSA models were trained on 16 32GB NVIDIA V100 GPUs for 10 days and saw ca. 3e9 tokens over 55,000 training steps.

Baseline models To enable direct comparison, the left-to-right autoregressive (LRAR) and CARP baselines were trained with the same CNN architectures on the same dataset as EvoDiff sequence models. For LRAR, the convolution modules have a causal mask to prevent information leakage. For additional MLM baselines, sequences were sampled from the protein MLMs

ESM-1b (*58*) and ESM-2 (*30*), which were trained on different releases of UniRef50. ESM-1b and ESM-2 both generated many "unknown" amino acids (`X`); performance was improved results by manually setting the logits for `X` to `inf`. Sequences were sampled from MLMs by treating the MLM as an OADMs and beginning from an all-mask state. For the structure-based diffusion baselines, sequences were obtained from FoldingDiff (*8*) and RFdiffusion (*10*) by first unconditionally generating structures and then using ESM-IF (*33*) to design their sequences. For MSA baselines, new query sequences were generated from ESM-MSA (*28*) by treating it as an OADM and sampling from an all-mask starting query sequence. CCMgen (*39*) with default parameters was used to train and generate from Potts models of validation MSAs from OpenFold.

Computation of test-set perplexities Perplexity was calculated by uniformly sampling a timestep for each test sequence, corrupting the sequence according to each diffusion model, predicting the sequence x_0 at $t = 0$ by passing inputs once through each trained model, and then computing the perplexity. For D3PM models, the perplexity is:

$$\text{Perp}_{\text{D3PM}} = \mathbb{E}_{t\sim\mathcal{U}(1,\ldots,T)} \exp\left(-\frac{1}{L}\sum_{i}^{L} \log p_\theta(x_0|x_t)\right) \tag{11}$$

For OADMs, the perplexity is:

$$\text{Perp}_{\text{OADM}} = \cdot\mathbb{E}_{t\sim\mathcal{U}(1,\ldots,L)}\mathbb{E}_{\sigma\sim\mathcal{U}(S_D)} \exp\left[\frac{-1}{L-t+1}\sum_{k\in\sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)})\right] \tag{12}$$

To enable model comparison, perplexities for MLMs (CARP, ESM-1b, ESM-2) were computed as if they are OADMs.

And for LRAR models, the perplexity is:

$$\text{Perp}_{\text{LRAR}} = \exp\left[-\sum_{t=1}^{L} \log p(x_t|\boldsymbol{x}_{<t})\right] \tag{13}$$

Calculated D3PM perplexities were on average higher as $t \to T$ and lower as $t \to 1$, and masked perplexities were similarly higher for a greater number of masked tokens per sequence, i.e., as $t \to L_{masked}$ (**Fig. S1, S2**). Lower perplexities indicated improved performance and generalization capacity.

Evaluation of structural plausibility The structural plausibility pipeline (**Fig. 2A**) evaluates both the foldability and self-consistency of a given sequence. Foldability was evaluated by averaging the per-residue confidence score, reported as pLDDT by OmegaFold, across the entire sequence. Sequence self-consistency, denoted scPerpelxity, describes how likely the generated sequence is to correspond to the predicted structure. Self-consistency was measured by taking structures predicted for a sequence from OmegaFold, running them through ESM-IF, and calculating the perplexity between the ESM-IF predicted-sequence and the original generated sequence.

The novelty of generated sequences was evaluated relative to training data seen by the model, by computing the Hamming distance between each generated sequence and every training-set sequence of the same sequence length. The minimum of these Hamming distances, representing the closest sequence seen by the model during training, was reported for each sequence.

Computation of functional and structural features To evaluate sequence coverage, ProtT5 embeddings were computed for each of 1,000 generated protein sequences and 10,000 sequences sampled from the test set using the Tools from Protein Prediction for Interpretation of Hallucinated Proteins (PPIHP) package (*69*). The resulting distributions of sequence embeddings (i.e., representing the corresponding distributions of sequences) were compared via the

Fréchet ProtT5 distance (FPD),

$$\text{FPD} = ||\mu_{test} - \mu_{gen}||^2 + \text{Tr}(C_{test} + C_{gen} - 2\sqrt{C_{test}C_{gen}}) \tag{14}$$

where, given the embedding space feature vectors for the test and generated distributions, μ is the feature-wise mean for each set of sequences, C is the respective covariance matrix, and Tr refers to the trace linear algebra operation, defined as the sum of the elements along the main diagonal of a square matrix. Embeddings were visualized in 2D via uniform manifold approximation and projection (UMAP), fit to the test data and with `n_neighbors`=25. The number of neighbors hyperparameter was selected to favor local similarities in place of global ones, in order to appropriately visualize the corresponding differences in embedding space and FPD measured for each model.

Structural features of generated sequences were evaluated via the ProtTrans (*37*) CNN predictor model to assign a 3-state secondary structure definition from DSSP (helix, strand, or other) to each residue in a protein. The fraction of predicted 'helix', 'strand', or 'other' was computed (the three values sum to 1 per sequence). The resulting multivariate distributions of secondary structure features (computed over 1000 generated or natural sequences) were visualized via kernel density estimation. The KL divergence between the mean values across the 3-state predictions for the generated and test sets was used to quantitatively evaluate the distribution of secondary-structures assigned for each model.

Evolution-guided generation with EvoDiff-MSA Starting with either a random or Max-Hamming subsampled MSA, new query sequences were generated by sampling from an all-mask starting query sequence. The generated query sequence was evaluated relative to the corresponding original query sequence using the same tools and workflow described in *Evaluation of structural plausibility*. Each generated sequence was additionally evaluated for similarity relative to its reference MSA, which is comprised of a query sequence and alignment

sequences. The % similarity of each generated sequence relative to its parent MSA was computed as the maximum % similarity over all sequences in the original MSA. Specifically, for a pair of sequences, the % similarity was computed by calculating the number of shared residue identities (accounting for both amino-acid identity and position index in the sequence), and for a given generated sequence the maximum value of these % similarities was determined. Across generated sequences both the CDF and mean of maximum % similarity were reported. Generated sequences were additionally evaluated for structural similarity relative to their original query sequences. Structures were predicted for each of the generated query sequences and the original query sequences using OmegaFold. Structural similarity was measured via the template modeling score (TM-score) (*70*) for the two predicted structures following structural alignment:

$$\text{TM-score} = \max \left[\frac{1}{L_{\text{gen}}} \sum_{i}^{L_{\text{common}}} \frac{1}{1 + \left(\frac{d_i}{d_0(L_{\text{true}})} \right)^2} \right] \tag{15}$$

where L_{gen} is the length of the generated query sequence; L_{common} is the number of shared residues; d_i is the distance between the i^{th} pair of residues; L_{true} is the length of the true query sequence; and $d_0(L_{\text{true}}) = 1.24\sqrt[3]{L_{\text{true}} - 15} - 1.8$ is a distance scale for normalization.

Generation of intrinsically disordered regions (IDRs) IDR generation and analysis leveraged a publicly available dataset of 15,996 human IDRs and their orthologs (*46*). This dataset was generated by running SPOT-Disorder v1 (*71*) on the human proteome and applying the predicted IDR positions to an MSA of likely-similar-function orthologs (determined using an evolutionary distance heuristic), curated from the larger set of orthologs contained in the OMA database (*72*). The resulting dataset only contained IDRs, not full protein sequences, and thus IDR sequences were mapped back to the MSAs of full protein sequences in OMA in order to provide context about the sequence regions surrounding the IDRs.

For input to EvoDiff models, the full sequence of an IDR-containing human protein was

treated as the query sequence, and a corresponding MSA was constructed by subsampling 63 other sequences from all the query's orthologs. All sequences were subsampled to 512 residues in length, with the following criteria maintained. Subsampling criteria were that the subsampled query sequence contain at least 1 IDR, and that the total IDR region was less than half the total length of the subsampled sequence ($L_{\text{IDR}} \leq 256$). For IDR generation from EvoDiff-Seq, the query sequence with the IDR region masked was provided as the only input to EvoDiff-Seq, which then generated new residues for the masked region (i.e., the region corresponding to the true IDR). For IDR generation from EvoDiff-MSA, the query sequence with the IDR region masked, aligned to the rest of the MSA, was provided as input to EvoDiff-MSA, which then generated new residues for the masked region.

The resulting generations, containing putative IDRs, were input to DR-BERT, a protein language model fine-tuned for disorder prediction (*47*), to obtain per-residue disorder scores ranging from 0-1 (less to more disordered). A single-sequence IDR predictor (DR-BERT) was used in place of MSA-based IDR scoring methods, because of an observed bias towards higher disorder scores with MSA-based methods – e.g., random uniform sampling of residues in the masked query positions still resulted in a prediction of disorder given the presence of the orthologs in the alignment. Disorder scores for true IDRs, generated IDRs, scrambled IDRs, and randomly generated IDRs were computed to evaluate the performance of DR-BERT predictions. The randomly-sampled baseline was constructed by randomly sampling amino acids over an IDR region; the scrambled baseline was constructed by shuffling the existing amino acids over an IDR region into a scrambled permutation. In all cases (true IDRs, generated IDRs, scrambled and random baselines), the entire protein sequence was input to DR-BERT for scoring. Since DR-BERT is for single-sequences, for putative IDRs generated by EvoDiff-MSA, the entire query sequence was inputted into DR-BERT, with $<$ `GAP` $>$ tokens eliminated, to obtain per-residue disorder scores. Lastly, a direct comparison between the original IDR and

the generated putative IDR was conducted by calculating the % sequence similarity between the fraction of shared residues between the two IDR regions.

Motif scaffolding Scaffolding performance was evaluated on a recently published benchmark (*10*) of 25 scaffolding problems across 17 unique proteins.

In our scaffolding benchmark, each unique protein was treated as 1 example, for a total of 17 unique scaffolding examples. 100 samples were generated for each unique scaffolding example. For proteins 6E6R, 6EXZ, 7MRX, and 5TRV, which were the 4 examples evaluated at 3 different scaffolding lengths in RFdiffusion (*10*), the number of successes across these three different scaffolding lengths were averaged to facilitate comparisons between RFDiffusion and EvoDiff.

To generate a scaffold with EvoDiff-Seq, a scaffold length between 50-100 residues (exclusive of the motif) was sampled uniformly; the motif was placed randomly within the length; and scaffold residues were generated from EvoDiff-Seq conditioned on the provided motif residues. In this approach, on average, protein sequences generated by EvoDiff-Seq were longer (between 45 and 194 residues in length) than those inverse-folded from structures generated by RFdiffusion, which range from 30-152 residues in total length inclusive of the length of the motif.

For scaffolding with EvoDiff-MSA, MSAs for each sequence corresponding to the original PDB structure were generated using the tools from AlphaFold (*73*) and then subsampled to 64 sequences, and a maximum of 150 residues in length, where the original sequence obtained from the PDB crystal structure was assigned as the query sequence. In cases where the scaffolding examples were shorter than 150 residues, sequences were padded with a $<$ GAP $>$ token, to allow EvoDiff to generate longer-scaffolds. Sequences generated by EvoDiff-MSA were between 56 and 150 residues in length, inclusive of the motif and scafffold. For each scaffold-

ing example, a common set of 100 subsampled MSAs, where 50 were randomly subsampled and 50 were subsampled via MaxHamming, was used commonly across EvoDiff-MSA (Max), EvoDiff-MSA (Random), and ESM-MSA. That is, an individual generation trial for each model corresponded to a unique MSA from the common set of 100 MSAs constructed for a scaffolding example. At inference time, all non-motif residues in the query sequence were masked, and new residues in these locations were generated by EvoDiff-MSA.

OmegaFold was used to predict structures corresponding to sequences generated by EvoDiff. A generation was counted as 'successful' if its predicted structure had a pLDDT ≥ 70 and a motifRMSD $\leq 1.0\text{Å}$ relative to the original motif crystal structure. Note that these success criteria are cutoffs proposed by structure-based models (*10*) and adopted here to facilitate comparison. The motifRMSD was computed as the RMSD between the alpha-carbons of the motif in the original crystal structure and the predicted structure for the scaffolded motif.

Supplementary Information for:

Protein generation with evolutionary diffusion: sequence is all you need

Sarah Alamdari[1], Nitya Thakkar[2,†], Rianne van den Berg[3], Alex X. Lu[1], Nicolo Fusi[1], Ava P. Amini[1], Kevin K. Yang[1,*]

[1]Microsoft Research, Cambridge, MA, USA
[2]Brown University, Providence, RI, USA
[3]Microsoft Research AI4Science, Amsterdam, Netherlands
[†]Work done principally during an internship at Microsoft Research New England
[*]To whom correspondence should be addressed; E-mail: yang.kevin@microsoft.com

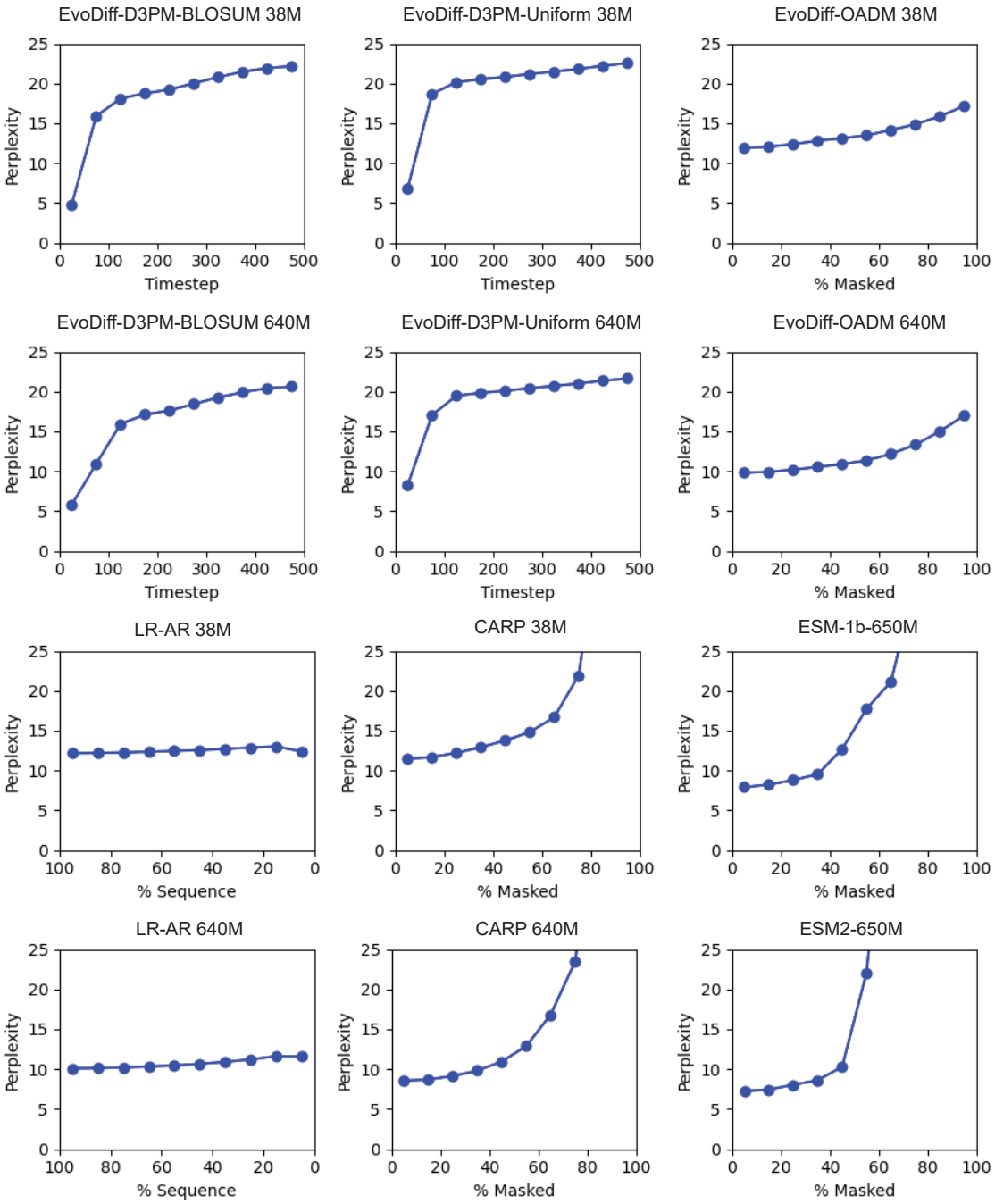

Figure S1: **Perplexity as a function of corruption step for EvoDiff sequence models.** Test-set perplexities at sampled intervals of the degree of corruption, specifically the diffusion timestep for D3PM models, the fraction of masked residues for OADM and masked language models, and the fraction of evaluated sequence for LRAR models. Intervals reflect evenly spaced windows of 50 timesteps for D3PM models or 10% masking for masked models.

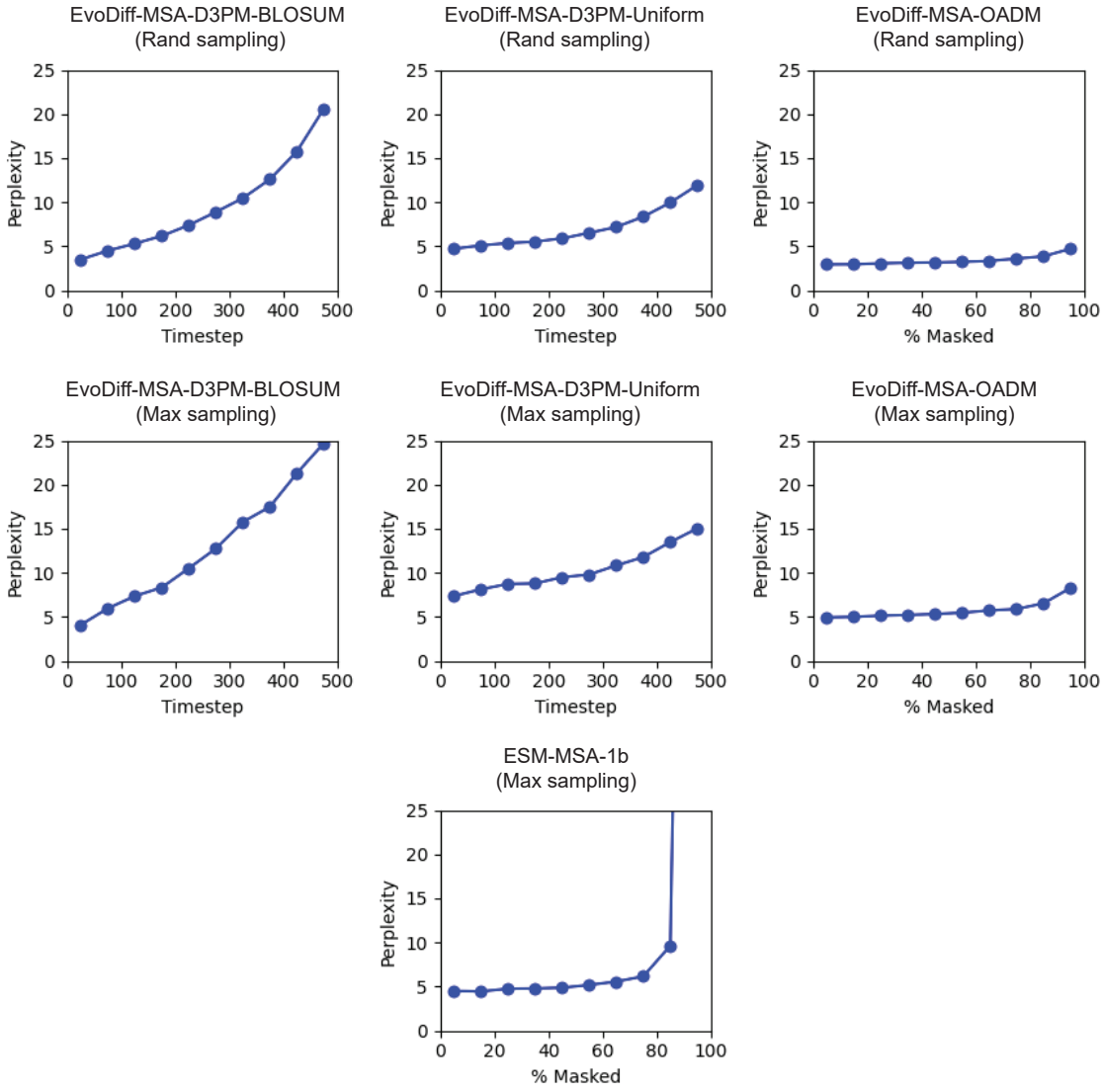

Figure S2: **Perplexity as a function of corruption step for EvoDiff MSA models.** Test-set MSA perplexities at sampled intervals of the degree of corruption, specifically the diffusion timestep for D3PM models and the fraction of masked residues for OADM and ESM models. The test-set evaluated for each model was sampled using the same sampling scheme assigned during training. Intervals reflect evenly spaced windows of 50 timesteps for D3PM models or 10% masking for masked models.

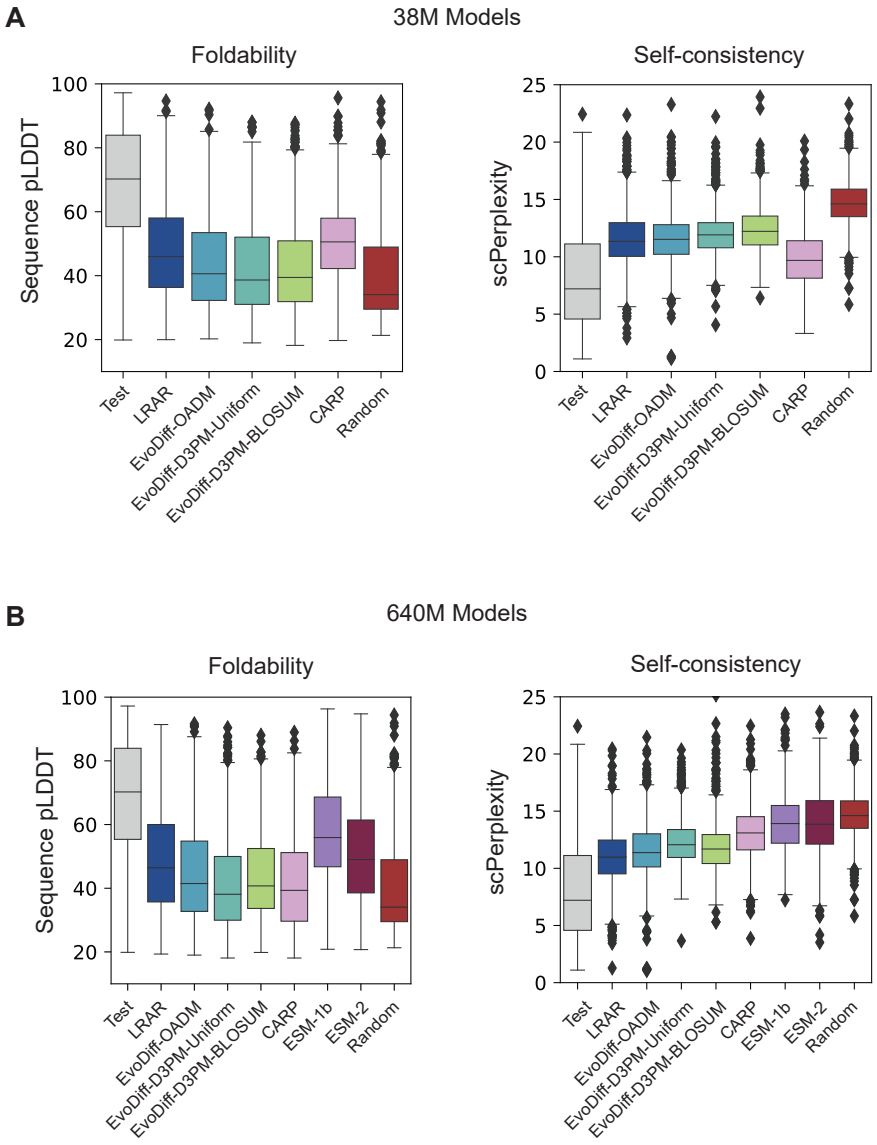

Figure S3: **Summary statistics for structural plausibility metrics for sequence models. (A-B)** Distribution of pLDDT and scPerplexity metrics for sequences from the test set, 38M parameter EvoDiff and baseline models (A), and 640M parameter EvoDiff and baseline models (B) (n=1000 sequences per model). Test and Random baselines are reproduced in (A) and (B) for reference.

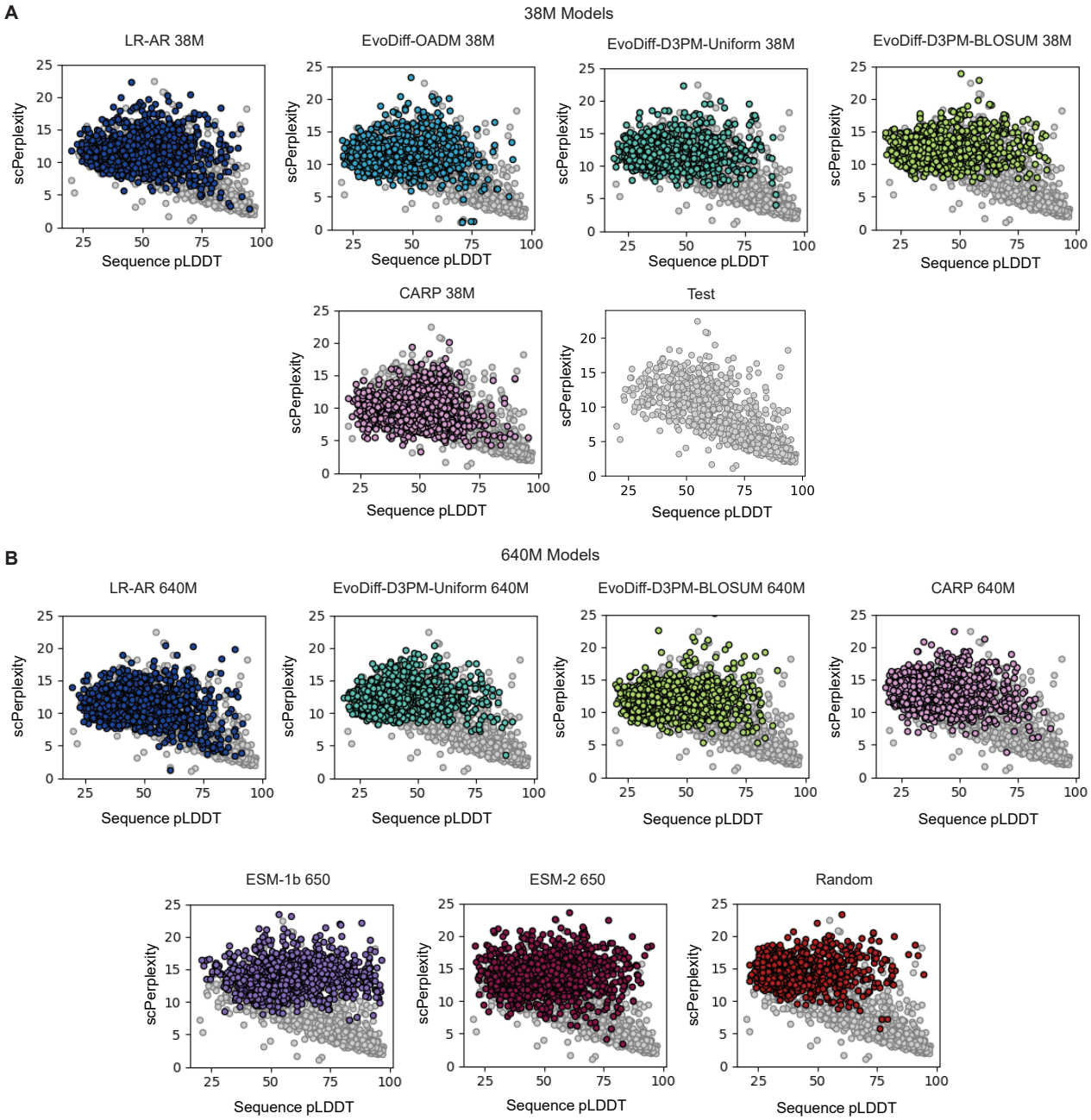

Figure S4: **Sequence pLDDT versus self-consistency perplexity for EvoDiff sequence models. (A-B)** Results for sequences from 38M parameter EvoDiff, baseline models, and test data plotted alone (A), and 640M parameter EvoDiff and baseline models (B) (various colors, n=1000), except for EvoDiff-OADM-640M (EvoDiff-Seq, shown in Fig. 2C), relative to sequences from the test set (grey, n=1000). The self-consistency perplexity (ESM-IF Perplexity) is computed using sequences inverse-folded by ESM-IF.

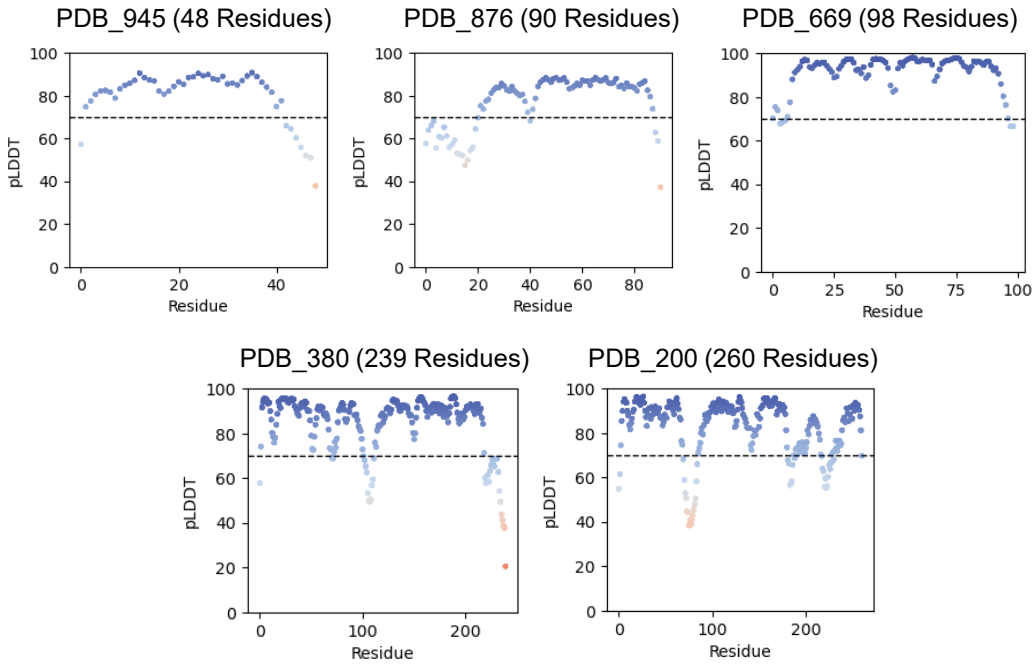

Figure S5: **Per-residue pLDDT for representative proteins generated by EvoDiff-Seq.** pLDDT scores computed based on the OmegaFold predicted structures, for individual residues in representative high-fidelity generations from EvoDiff-Seq (Fig. 2E). Points are colored by pLDDT (0-100, red to blue).

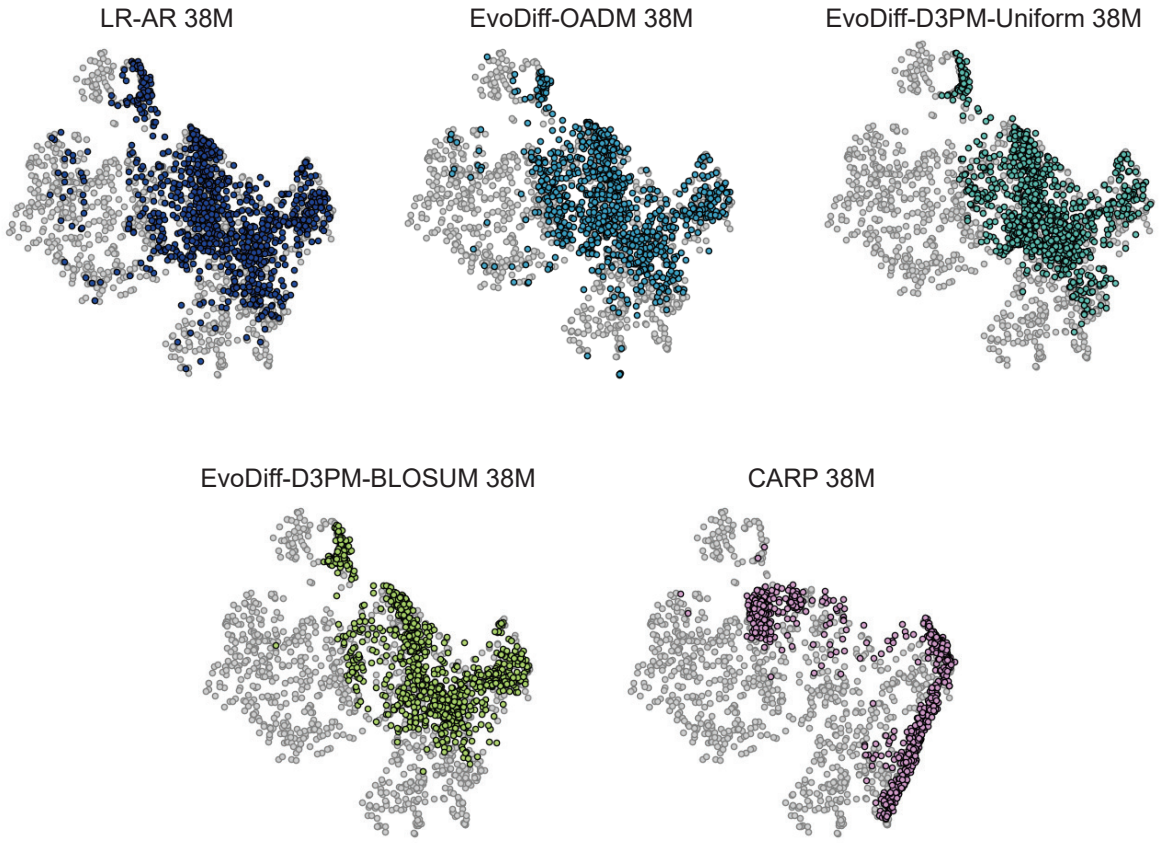

Figure S6: **Coverage of sequence and functional space for generated distributions from 38M parameter EvoDiff sequence models and baselines.** UMAP of ProtT5 embeddings, annotated with FPD, of natural sequences from test set (grey, n= 1000 plotted) and of generated sequences from EvoDiff 38M parameter models and baselines (various colors, n=1000).

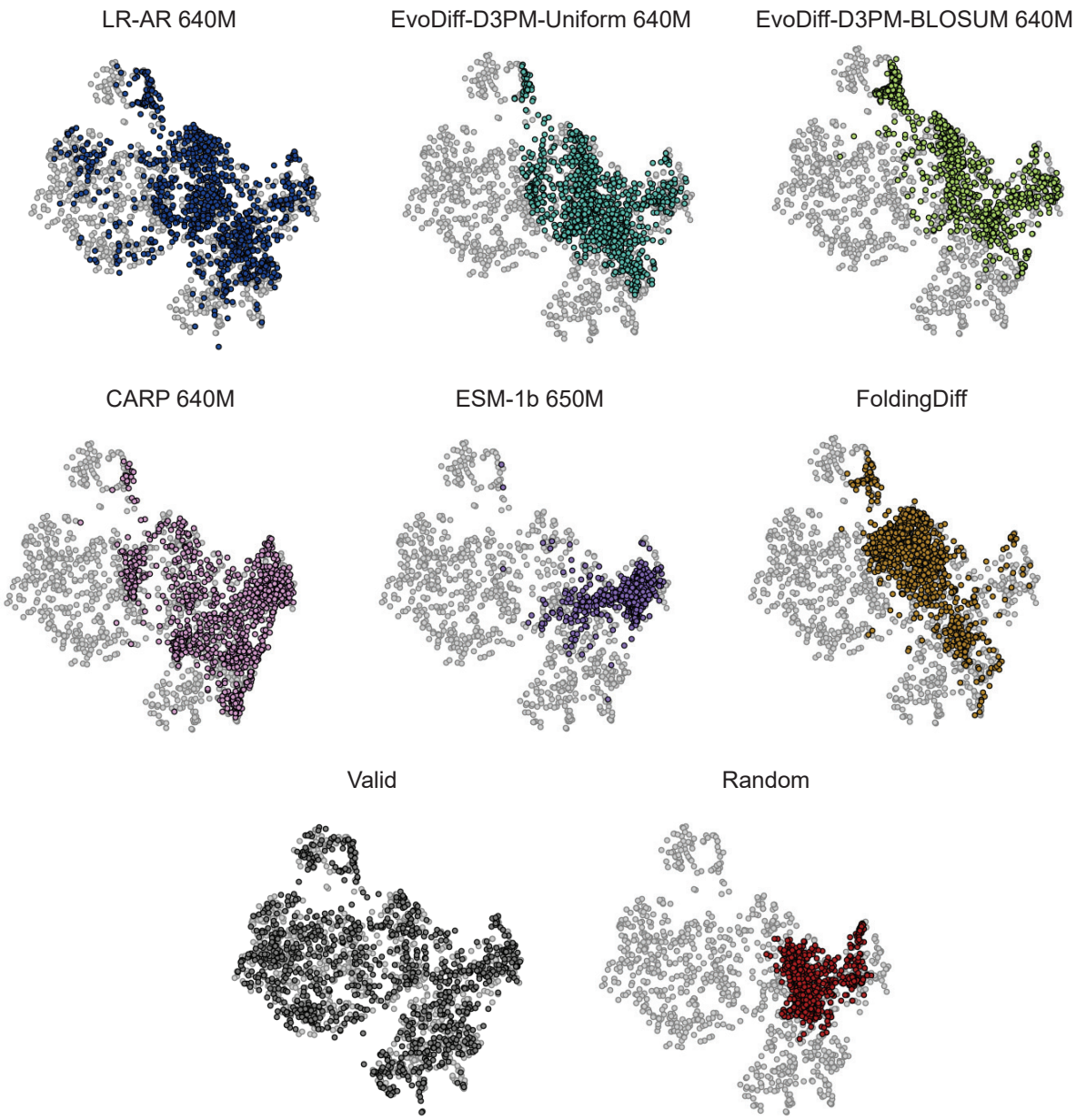

Figure S7: **Coverage of sequence and functional space for generated distributions from 640M parameter EvoDiff sequence models and baselines.** UMAP of ProtT5 embeddings, annotated with FPD, of natural sequences from test set (grey, n= 1000) and of generated sequences from EvoDiff 640M parameter models and baselines (various colors, n=1000). A visualization of sequences from the validation set (dark grey, n=1000) is included for reference. The visualization for the 640M OADM model is excluded due to inclusion in Fig. 3A.

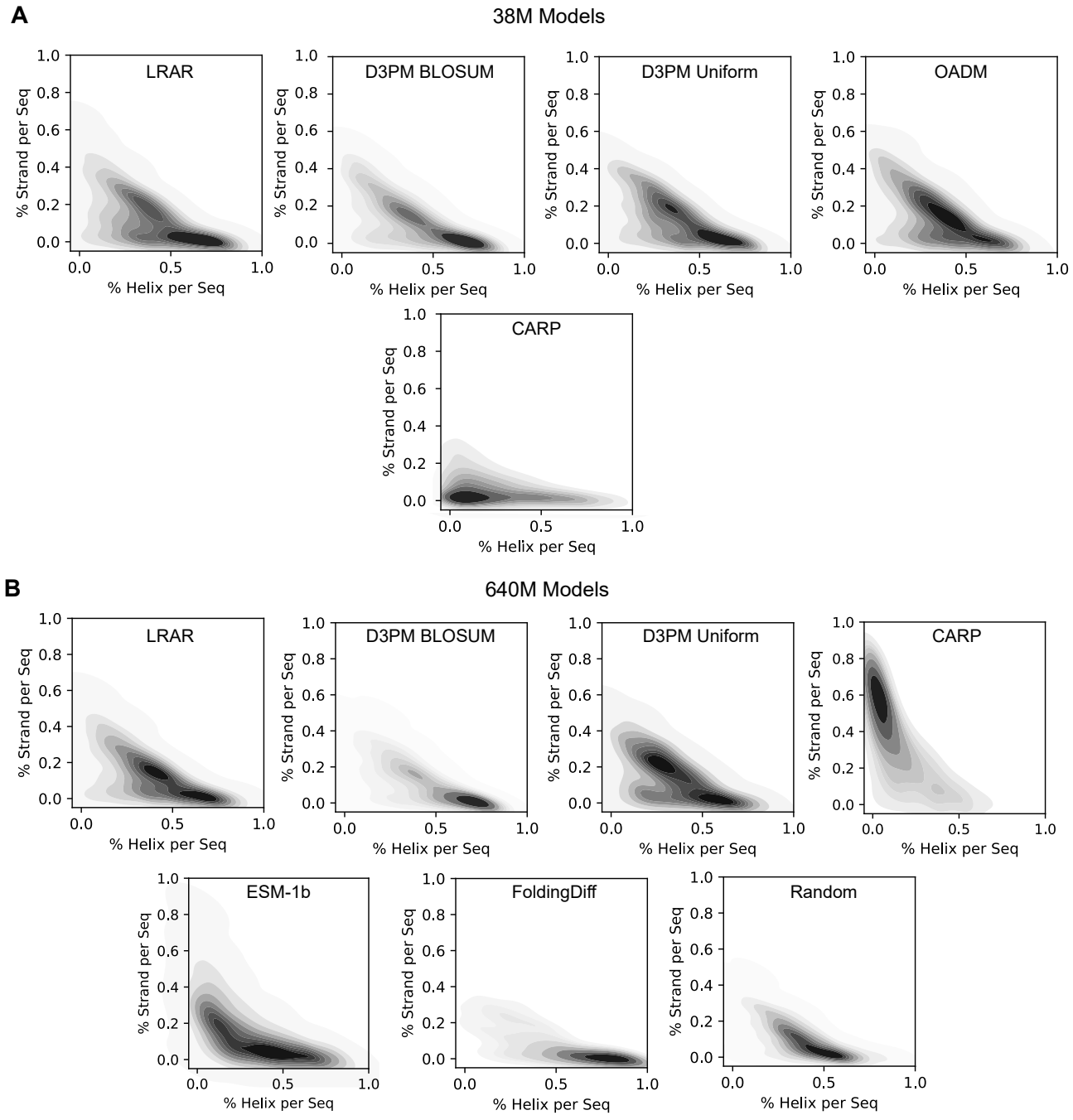

Figure S8: **Structural features in generated sequences from all sequence models. (A-B)** Multivariate distributions of helix and strand features in sequences from 38M (A) and 640M (B) parameter models, and baselines based on DSSP 3-state predictions and annotated with the KL divergence relative to the test set (n=1000 samples from each model). In (B), the distribution for the 640M OADM model is excluded (see Fig. 3B); the distribution for random sequences (n=1000) is provided as reference.

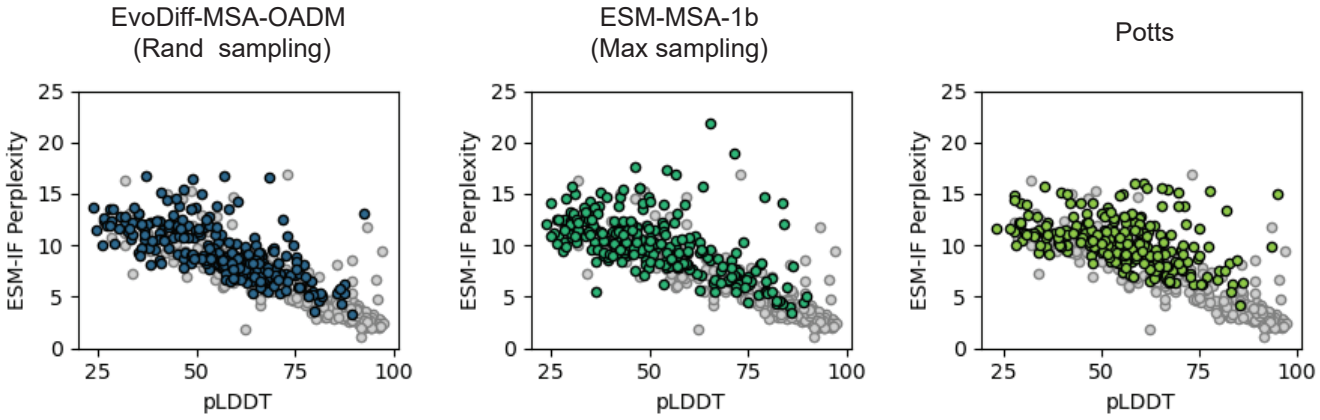

Figure S9: **Sequence pLDDT versus scPerplexity for EvoDiff MSA models,** for sequences from the validation set (grey, n=250) and evaluated MSA models (various colors, n=250), except for EvoDiff-OADM-MSA-Max (EvoDiff-MSA, shown in Fig. 4F).

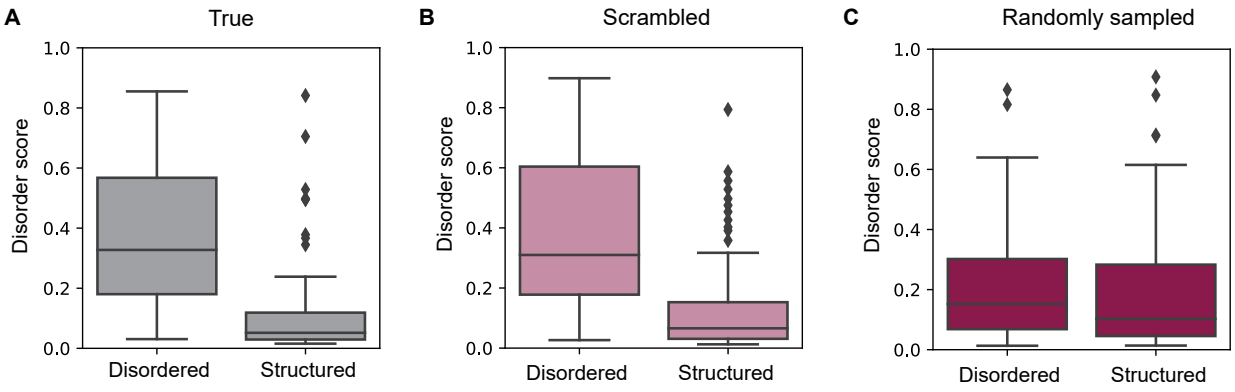

Figure S10: **Baseline performance of DR-BERT evaluator. (A-C)** Distributions of DR-BERT predicted disorder scores across disordered and structured regions for sequences with true (A), scrambled (B), and randomly sampled (C) IDRs (n=100).

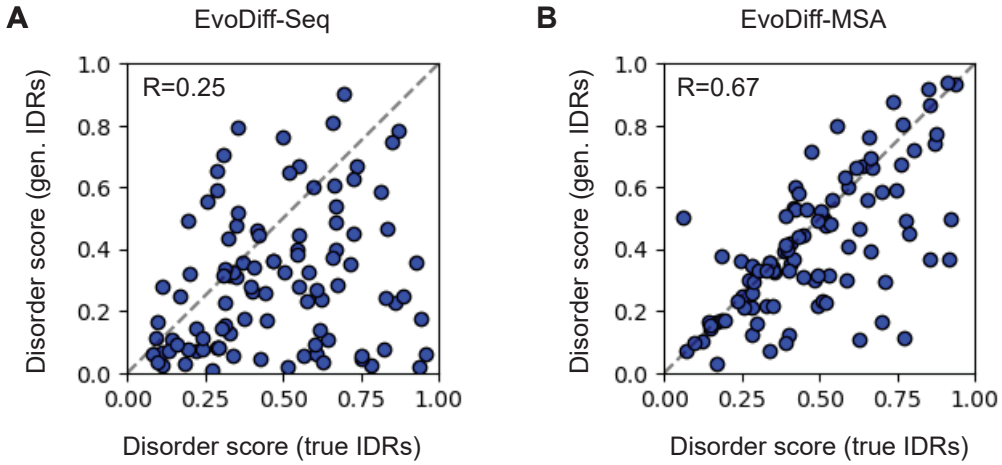

Figure S11: **Performance of DR-BERT evaluator on disorder regions. (A-B)** Disorder scores predicted by DR-BERT for true (x-axis) vs. generated (y-axis) IDRs for the same given sequence, for generations from EvoDiff-Seq (A) and EvoDiff-MSA (B). Each dot represents an individual IDR (n=100). The Pearson R is given for each of EvoDiff-Seq and EvoDiff-MSA.

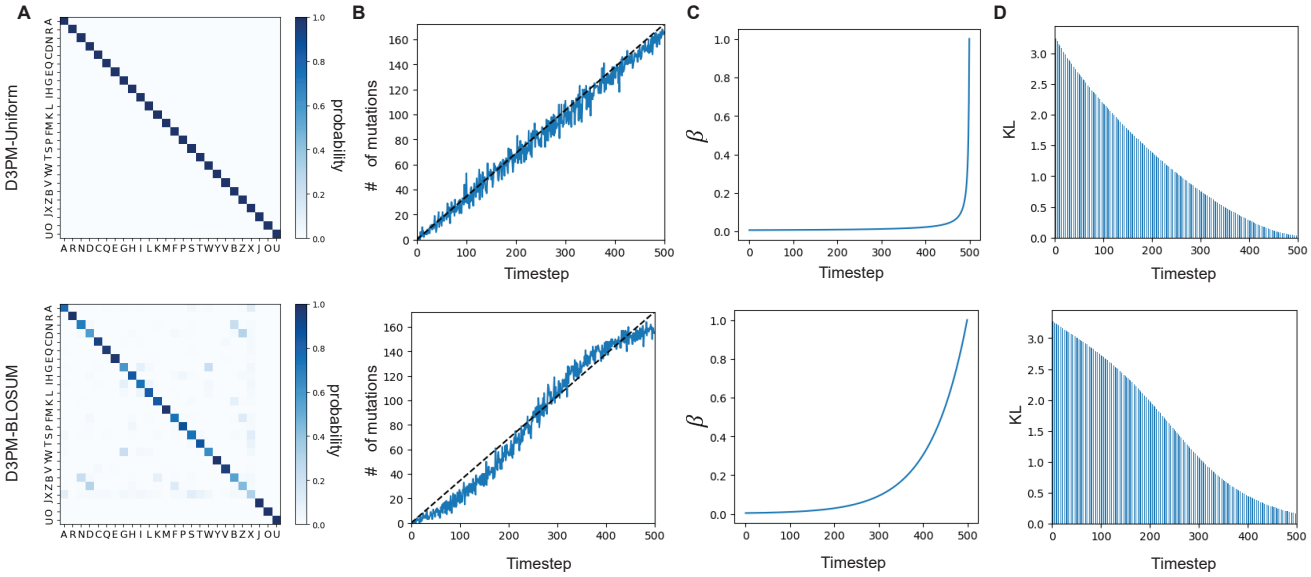

Figure S12: **Details of EvoDiff-D3PM corruption schemes.** The top and bottom rows correspond to EvoDiff-D3PM-Uniform and EvoDiff-D3PM-BLOSUM, respectively. **(A)** Visualization of EvoDiff-D3PM transition matrices. **(B)** Evolution of the number of mutations accrued as a function of the diffusion timestep t for a sample input. **(C)** Evolution of β as a function of the diffusion timestep t. **(D)** Evolution of $D_{KL}\left[q(x_t|x_0)\|p(x_T)\right]$ as a function of the diffusion timestep t, indicating convergence to a uniform stationary distribution at $t = 500$ as D_{KL} approaches zero.

Table S1: **Performance of EvoDiff sequence models**. The reconstruction KL (Recon KL) was calculated between the distribution of amino acids in the test set and in generated samples (n=1000). The perplexity was computed on 25k samples from the test set. The minimum Hamming distance to any train sequence of the same length (Hamming) is reported for each model as the mean $\pm$ standard deviation over the generated samples. Fréchet ProtT5 distance (FPD) was calculated between the test set and generated samples. The secondary structure KL (SS KL) was calculated between the means of the predicted secondary structures of the test and generated samples.

Model	parameters	Recon KL	perplexity	Hamming	FPD	SS KL
Test	-	9.92e-4[1]	-	0.0039[2]	0.10[1]	1.37e-5[1]
D3PM BLOSUM	38M	1.77e-2	17.16	0.83 $\pm$ 0.05	1.42	3.30e-5
D3PM Uniform	38M	1.48e-3	18.82	0.83 $\pm$ 0.05	1.31	3.73e-5
OADM	38M	1.11e-3	14.61	0.83 $\pm$ 0.07	0.92	1.61e-4
D3PM BLOSUM	640M	3.73e-2	15.74	0.83 $\pm$ 0.05	1.53	4.96e-4
D3PM Uniform	640M	2.90e-3	18.47	0.83 $\pm$ 0.05	1.35	2.13e-4
OADM	640M	1.26e-3	13.05	0.83 $\pm$ 0.08	0.88	1.48e-4
LRAR	38M	7.90e-4	12.38	0.82 $\pm$ 0.06	0.86	1.61e-4
CARP	38M	5.71e-1	25.13	0.74 $\pm$ 0.07	6.30	2.72e-3
LRAR	640M	7.01e-4	10.41	0.83 $\pm$ 0.06	0.63	1.76e-5
CARP	640M	3.56e-1	31.77	0.84 $\pm$ 0.05	1.78	5.03e-3
ESM-1b [3]	650M	4.91e-1	53.49	0.83 $\pm$ 0.06	6.67	5.48e-4
ESM-2 [3]	650M	5.00e-1	68.39	0.84 $\pm$ 0.06	6.79	3.05e-3
FoldingDiff [4]	14M	5.49e-2	-	-	1.64	1.76e-3
RFdiffusion [5]	60M	7.19e-2	-	-	1.96	5.98e-3
Random	-	1.65e-1	20	0.85 $\pm$ 0.04	3.16	1.90e-4

[1] Calculated between the test set and validation set.

[2] Reported value is the minimum Hamming distance between any two natural sequences of the same length in UniRef50.

[3] Due to model constraints, the maximum sequence length sampled was 1022.

[4] For the FoldingDiff baseline, 1000 structures generated by FoldingDiff were randomly selected, and the corresponding 1000 inferred sequences were inverse-folded using ESM-IF. These sequences are between lengths of 50 and 128 residues.

[5] For the RFdiffusion baseline, 1000 structures were generated corresponding to the UniRef train distribution length, and 1000 corresponding sequences were inverse-folded using ESM-IF.

Table S2: **Validation-set perplexities for EvoDiff MSA models.** The perplexity is calculated based on the ability of each model to reconstruct a subsampled MSA from the validation set. "Max Perplexity" and "Rand. Perplexity" indicate MaxHamming and Random subsampling, respectively, for construction of the validation MSA.

Corruption	Subsampling	Params	Max Perplexity	Rand. Perplexity
D3PM BLOSUM	Random	100M	11.35	8.31
D3PM BLOSUM	Max	100M	10.98	7.61
D3PM Uniform	Random	100M	10.14	6.77
D3PM Uniform	Max	100M	10.06	6.66
OADM	Random	100M	6.05	3.64
OADM	Max	100M	6.14	3.60
ESM-MSA-1b	Max	100M	11.20	5.89

Table S3: **Structural plausibility metrics for EvoDiff sequence models and baselines.** Metrics are reported as the mean ± standard deviation for 1000 generated samples for each model.

Model	Params	ESM-IF scPerplexity	ProteinMPNN scPerplexity	OmegaFold pLDDT
Test	-	8.04 ± 4.04	3.09 ± 0.63	68.25 ± 17.85
D3PM Blosum	38M	12.38 ± 2.06	3.80 ± 0.49	42.76 ± 14.55
D3PM Uniform	38M	12.03 ± 2.04	3.77 ± 0.50	42.37 ± 14.39
OADM	38M	11.61 ± 2.38	3.72 ± 0.50	43.78 ± 14.18
D3PM Blosum	640M	11.86 ± 2.21	3.73 ± 0.48	44.14 ± 13.80
D3PM Uniform	640M	12.29 ± 2.05	3.78 ± 0.49	41.65 ± 14.32
OADM	640M	11.53 ± 2.50	3.71 ± 0.52	44.46 ± 14.62
LRAR	38M	11.61 ± 2.38	3.64 ± 0.56	48.26 ± 14.87
CARP	38M	9.68 ± 2.56	3.66 ± 0.62	50.79 ± 12.06
LRAR	640M	10.99 ± 2.63	3.59 ± 0.54	48.71 ± 15.47
CARP	640M	14.13 ± 2.42	4.05 ± 0.52	41.56 ± 14.35
ESM-1b	650M	13.90 ± 2.44	3.47 ± 0.68	58.07 ± 15.64
ESM-2	650M	14.02 ± 2.87	3.58 ± 0.69	50.70 ± 15.67
Random	-	14.68 ± 1.97	3.96 ± 0.50	39.97 ± 14.05

Table S4: **Performance of EvoDiff MSA models in generating query sequences conditioned on MSAs.** Metrics are reported as the mean ± standard deviation over 250 generated samples for each model.

Model	scPerplexity	pLDDT	Seq. similarity	TM score
Valid	5.93 ± 3.19	73.99 ± 17.80	14.58 ± 21.64 [1]	-
OADM (Rand) - Rand MSA	9.41 ± 2.61	55.99 ± 14.75	6.13 ± 9.88	0.49 ± 0.23
OADM (Max) - Max MSA	9.38 ± 2.57	57.08 ± 16.01	6.74 ± 11.00	0.50 ± 0.23
OADM (Max) - Rand MSA	9.59 ± 2.69	54.95 ± 16.83	6.55 ± 10.49	0.46 ± 0.23
ESM-MSA-1b	10.05 ± 2.92	51.64 ± 16.54	7.13 ± 11.60	0.40 ± 0.23
Potts	10.34 ± 2.26	55.46 ± 13.82	12.01 ± 17.19	0.17 ± 0.10

[1] Sequence similarity is calculated between the original query sequence and all the sequences in the MSA.

Table S5: **Scaffolding performance of EvoDiff-Seq.** Number of scaffolding successes out of 100 generations for RFdiffusion, EvoDiff-Seq, the LRAR baseline, the CARP baseline, and randomly sampled scaffolds (Random), for each of 17 scaffolding problems. The bottom row contains the total number of successful scaffolds generated per model.

PDB	RFdiffusion	EvoDiff-Seq	LRAR	CARP	Random
1BCF	100	24	0	4	0
6E6R	71	16	7	3	1
2KL8	88	0	1	1	0
6EXZ	42	0	0	0	0
1YCR	74	13	12	10	7
6VW1	69	1	0	0	0
4JHW	0	0	0	0	0
5TPN	61	0	0	0	0
4ZYP	40	0	0	0	0
3IXT	25	23	22	13	7
7MRX	7	0	0	0	0
1PRW	8	68	70	54	5
5IUS	2	0	0	0	0
5YUI	0	4	0	0	0
5WN9	0	0	0	0	2
1QJG	0	0	0	0	0
5TRV	22	0	0	0	0
Total	610	149	112	85	22

Table S6: **Scaffolding performance of EvoDiff-MSA.** Number of scaffolding successes out of 100 generations for RFdiffusion, EvoDiff-MSA (Max), EvoDiff-MSA (Random), and the ESM-MSA baseline, for each of 17 scaffolding problems. The bottom row contains the total number of successful scaffolds generated per model.

PDB	RFdiffusion	EvoDiff-MSA (Max)	EvoDiff-MSA (Random)	ESM-MSA
1BCF	100	100	98	99
6E6R	71	87	63	96
2KL8	88	11	31	42
6EXZ	42	86	87	73
1YCR	74	3	0	0
6VW1	69	4	3	4
4JHW	0	0	0	0
5TPN	61	0	0	0
4ZYP	40	0	0	0
3IXT	25	1	0	5
7MRX	7	72	68	66
1PRW	8	48	46	92
5IUS	2	3	1	7
5YUI	0	58	44	70
5WN9	0	0	0	0
1QJG	0	34	22	38
5TRV	22	15	12	12
Total	610	522	475	604