DESIGN IN THE DARK: LEARNING DEEP GENERATIVE MODELS FOR DE NOVO PROTEIN DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

The design of novel protein sequences is providing paths towards the development of novel therapeutics and materials. Generative modelling approaches to design are emerging and to date have required conditioning on 3D protein structure-derived information, and unconditional models of protein sequences have so far performed poorly. Thus, it is unknown if unconditional generative models can learn a distribution of sequences that captures structure information without it being explicitly provided, and so be of use in important tasks like *de novo* protein sequence design, where it is not possible to condition on structure. Here, we demonstrate that it is possible to use *unconditioned* generative models to produce realistic samples of protein sequences. We progressively grow a dataset of over half a million synthetic sequences for training autoregressive language models, using an iterative framework we call DARK. It begins by training an autoregressive model on an initial sample of synthetic sequences, sampling from it, and refining the samples thus generated, which are then used for subsequent rounds of training. Using the confidence measures provided by AlphaFold and other measures of sample quality, we show that our approach matches or exceeds the performance of prior methods that use weak conditioning on explicit structural information, and improves after each iteration of DARK. Crucially, the DARK framework and the trained models are entirely unsupervised; strong structural signal is an objective, but no model is ever conditioned on any specific structural state. The trained model indirectly learns to incorporate a structural signal into its learned sequence distribution, as this signal is strongly represented in the makeup of the training set at each step. Our work demonstrates a way of unconditionally sampling sequences and structures jointly, and in an unsupervised way.

1 INTRODUCTION

Generative modelling is beginning to be used for the task of designing protein molecules, a problem which offers potential solutions to a vast number of medical (Chevalier et al., 2017; Silva et al., 2019) and industrial challenges (King et al., 2012; Wang et al., 2021). The focus of protein design is to develop automatic methods for designing proteins that have pre-specified structural and functional attributes. Recent generative modelling studies have looked to improve on the high time and computational cost of contemporary design methods, which rely on Monte Carlo sampling and energy function-based physics simulations (Huang et al., 2016). A number of successful generative models have focused on the problem of sampling protein sequences conditioned on structural information (Strokach et al., 2020; Cao et al., 2021) and some have shown that unconditional sequence models perform poorly, suggesting they learn little regarding important sequence-structure relationships (Ingraham et al., 2019).

In this work, we show that autoregressive language models can learn unconditional distributions of protein sequences that incorporate latent 3D protein structure information and produce diverse samples with realistic predicted structures. We do so by developing Design in Areas of Restricted Knowledge (DARK), a framework for jointly learning unconditional generative sequence models and generating the synthetic sequences they're trained on. DARK iteratively grows a large dataset of over half a million synthetic sequences that are optimized to have a predicted structure that is highly likely under a supervised protein structure predictor. As the dataset increases in size, it enables training of progressively more performant deep generative models.

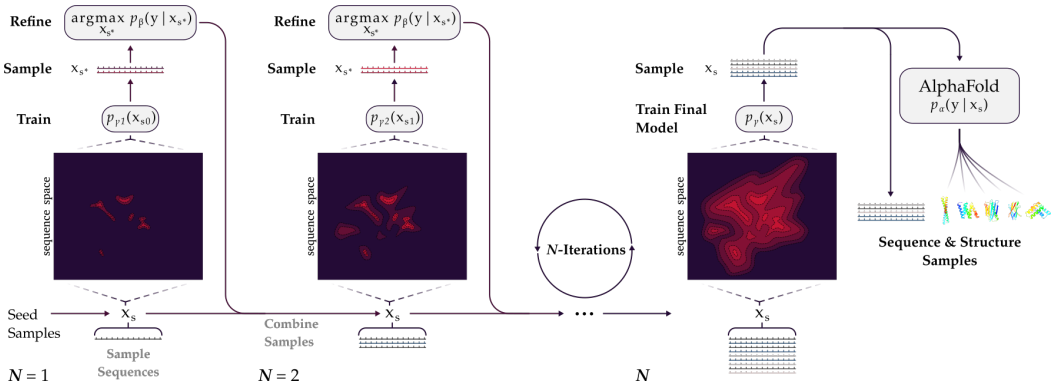

Figure 1: **Growing a synthetic dataset and training unconditional language models with DARK.** An illustrative overview of the procedure used by DARK to iteratively grow a training set of synthetic sequences that are used to train increasingly performant generative models. The final structure generation with AlphaFold for sampled sequences is also included.

A key benefit of learning these unconditional models is that by pairing them with highly accurate structure predictors, such as AlphaFold (Jumper et al., 2021; Tunyasuvunakool et al., 2021), they provide a way to unconditionally sample both protein sequence and structure jointly in a neural way. This in turn provides a fully learned way to perform *de novo* design, a strict type of design that uses little to no conditioning information, and looks to sample proteins from the huge 'dark' areas of the sequence space that are unexplored by nature (Taylor et al., 2009; Perdigão et al., 2015).

We rigorously evaluate our approach in several ways, most notably, by measuring the sample quality of DARK models using AlphaFold. Furthermore, we directly benchmark the likelihood performance of DARK models as well as provide several additional quantitative measures of sample diversity and quality. We find that DARK models unconditionally produce diverse sequence samples of a higher quality than equivalent models trained with natural sequences or models that use weak conditioning information.

In summary, we make the following contributions:

- We show that unconditional generative models can learn distributions of novel protein sequences that capture the important long range dependencies that relate protein sequences with their structures.
- We propose a novel framework, DARK, for learning deep generative models that produce diverse and high-quality protein sequence samples.
- We define and demonstrate several quantitative measures of sample quality for this task using pre-existing tools and state-of-the-art deep structure predictors, like AlphaFold.
- By pairing DARK models with models like AlphaFold, we show a fully neural way to unconditional sample both protein sequences and structures jointly.

2 Related Work

Here, we discuss related work applying generative models to protein design as well as those that have focused on *de novo* protein and broader protein design methods.

Generative sequence modelling in protein design There have been a variety of machine learning approaches (Killoran et al., 2017; Wang et al., 2018; Norn et al., 2021) including generative modelling (Anand & Huang, 2018; Sabban & Markovsky, 2020; Linder et al., 2020) in protein design and protein engineering (Yang et al., 2018; Sinai & Kelsic, 2020). Unconditional and conditional generative models have been explored in a variety of problems adjacent to protein design, such as variant prediction (Riesselman et al., 2018; Shin et al., 2021). Other examples are recent work using language models to learn representations for downstream supervised tasks (Alley et al., 2019; Bepler & Berger, 2019; Elnaggar et al., 2020; Rao et al., 2021).

Most relevant to the work presented here is the set of models that have been developed to explore design using generative models conditioned on structural information. Although a range of architectures have been used, these models vary primarily in the representations of structural information used for conditioning. Amongst others, models have conditioned on low information descriptions of protein folds (weak conditioning) (Greener et al., 2018; Karimi et al., 2020), secondary structure labels (Singer et al., 2021), and course descriptions of protein backbones (Ingraham et al., 2019; Strokach et al., 2020; Cao et al., 2021). Some methods train proteins of one structural or functional type, effectively conditioning on that specific type (Yu & Buehler, 2020; Shin et al., 2021). We highlight the recent work by Ingraham et al. (2019) as particularly relevant as it provides a valuable comparison between unconditioned and structure conditioned autoregressive language models. The authors show strong evidence of unconditioned models trained on only natural sequences performed close to random when evaluated on sequences with structures unlike those in the training set, suggesting they learned little general structure information.

Machine Learning in *de novo* protein design *De novo* design explores the difficult problem of design proteins with as little helping information as possible, as if 'from scratch' (Woolfson, 2021). In practice this entails designing a sequence and structure jointly, with as little conditioning information as possible. Generative modelling work so far has primarily focused on developing structure conditional models as described above. While very useful, these tackle a sub-problem of the full *de novo* design task by conditioning out structure. A highly relevant work that does considers structure and sequence is trDesign (Anishchenko et al., 2020), which we build on here. It samples sequences from structures prediction models by using simulated annealing to maximize the structure likelihood of random sequences.

Contemporary *de novo* protein design Although recent nonparametric methods have been develped (Zhou et al., 2020), by far the most historically successful *de novo* design approach is the Rosetta framework (Leaver-Fay et al., 2011), which is the recognized gold-standard (Huang et al., 2016). The Rosetta approach constitutes a complex computational pipeline that uses a variety of different protocols to iteratively sample viable structures and sequences. Importantly, it utilizes physics-based force-field simulations as well as Monte Carlo search methods (Vorobieva et al., 2021; Marcos et al., 2018) making it lengthy and computationally expensive. Significant human knowledge is also required due to the depth and intricacies of the Rosetta software suite. New techniques that can rapidly produce quality samples, such as the work presented here, offer the potential to greatly reduce these costs. This is especially important given the high costs and attrition rates of *in vitro* lab validation. For further discussion regarding the development of *de novo* design as a research focus, we point the reader to the recent review by Woolfson (2021), and more generally the reviews by Huang et al. (2016) and Korendovych & DeGrado (2020).

3 Learning unconditional generative sequence models

3.1 Background

Let $x \in V^L$ be a protein sequence of length L over a vocabulary V of discrete amino acids ($|V| = 20$). Each amino acid is textually represented by a one-letter code. Each sequence x maps to a corresponding molecular structure y in three-dimensional space. Most commonly, the structure is described by the 3D coordinates of each atom, however more coarse-grained representations are commonly used. Here, we use the discrete representation of structure known as distograms[1] (Senior et al., 2020).

Distograms represent a structure as a $[L, L, D]$ tensor, comprising D-dimensional one-hot encodings of a discretized distance measure between residue pairs. We do not consider the amino acid Cysteine for experimental reasons when generating synthetic sequences, however, for consistency we allow all trained models to predict it. For simplicity, we assume a fixed length of $L = 100$, however our approach easily generalizes to variable sequence lengths as it is based on autoregressively factorized probability models.

[1] A portmanteau of distance matrix and histogram.

3.2 DE NOVO DESIGN FROM A PROBABILISTIC PERSPECTIVE

De novo design can be framed as wanting to draw high probability samples from some ground-truth distribution over the joint sequence $\mathcal{X}$ and structure $\mathcal{Y}$ space. More accurately, we are interested in the set of sequences that are unlike those in nature $x_d \in \mathcal{X}$ with structures that are folded in some ordered and regular state $y_d \in \mathcal{Y}$. It is convenient to break this down, by Bayes, to $p(x_d, y_d) = p(y_d|x_d)p(x_d)$. The first term $p(y_d|x_d)$ can be thought of as protein structure prediction and so it is convenient to model it with some accurate proxy regression model $p_\beta(y_d|x_d)$, being a trained supervised structure predictor, with parameters $\beta \in B$. We refer to this proxy regression model as an *oracle*.

This leaves the prior $p(x_d)$ which is what we would like to model. A sufficient number of x_d is not directly available, so instead we aim to approximate it with a large number of high-quality samples $x_s \approx x_d$. In this work, we focus on modeling the prior with powerful autoregressive language models, and we aim to show that a sufficiently strong prior, combined with a state-of-the-art structure predictor, provides an efficient means to produce high quality sequence and structure samples without conditioning information.

3.3 THE CHALLENGES OF THE PROBLEM DOMAIN

A lack of training examples As stated previously, we seek to generate sufficient synthetic[2] samples, x_s, as x_d is not available in a sufficient quantity. Just over 100 *de novo* designed sequences (Woolfson, 2021) have lab-determined structures in the Protein Data Bank (PDB) (Burley et al., 2021). Although millions of natural sequences are available, they present several further issues that make leveraging them for training with a generative approach non-trivial at best. Firstly, the sequences we are interested in are by definition out-of-distribution (OOD) to natural sequences. Sequence databases are also a biased sample of the sequence space, largely being many small clusters of similar sequences, due to the effects of evolution. Taking these points into account, it is unclear how sequence samples distinct from natural sequences can be reliably generated from models trained on natural sequences. Indeed, using natural sequences in this way also raises questions regarding the extent to which the method can be called *'de novo'*.

Learning sequences with a strong structure signal We wish to learn distributions of sequences with strong signals for structure prediction. For the vast majority of known proteins, this signal is weak or mixed in with other types of information, such as functional constraints on amino acid types at certain sequence positions. Pre-existing *de novo* designed proteins can often be accurately predicted by models like AlphaFold, using only sequence, demonstrating that there do exist protein sequences with strong structure signals. Given the aforementioned lack of examples, finding a way to learn a distribution of sequences with these signals appears nontrivial. Recently, Anishchenko et al. (2020) demonstrated a way of 'hallucinating' *de novo* protein sequences from supervised structure predictors using a Monte Carlo sampling and optimization strategy, which the authors refer to as trDesign. Although slow, it demonstrates a way to generate unconditional sequence samples that have the criteria we desire. In the following section we describe how we leverage several concepts in trDesign to build DARK.

3.4 AN UNSUPERVISED DESIGN LOSS FUNCTION

As part of building DARK, we leverage several concepts proposed by the recent trDesign method, primarily the design loss function. trDesign demonstrates a method of sampling sequences for *de novo* design, which, like DARK, looks to maximize $p(y_d|x_d)p(x_d)$, but using the trRosetta structure predictor as an oracle (Yang et al., 2020). This is done by using simulated annealing on random sequences to optimize a combined proxy objective for both terms. $p(y_d|x_d)$ is approximated using Kullback-Leibler divergence (D_{KL}) between an oracle and a trained *background* network $p_\epsilon(y_d)$ which approximates a prior over ordered structures

$$\arg\max_{x_d} \log p_\beta(y_d|x_d) \approx \arg\max_{x_d} D_{\mathrm{KL}}\Big(p_\beta(y_d|x_d) \| p_\epsilon(y_d)\Big) \tag{1}$$

[2] All *de novo* designed sequences are inherently synthetic; we use this term to refer only to samples generated in this work.

The background network is trained by taking a model with the same architecture and dataset as $p_\beta(y_d|x_d)$ and training it with the sequence inputs replaced by Gaussian noise[3]. The final objective also includes a term for approximating a weak sequence prior by taking the negative D_{KL} between the sequence's normalized unigram frequencies $p(x_d^{uni})$ and the unigram frequencies of proteins with high resolution crystal structuresin the PDB, $p(x_{PDB})$. For simplicity, we refer to this objective as the Information Gain score (IG-score):

$$\text{IG-score} = \arg\max_{x_d} \left(D_{\mathrm{KL}}\Big(p_\beta(y_d|x_d) \| p_\epsilon(y_d)\Big) - D_{\mathrm{KL}}\Big(p(x_d^{uni}) \| p(x_{PDB})\Big)\right) \tag{2}$$

To use the IG-score with DARK we train a deep structure predictor and background network. We implement a version of the recent DMPfold2 model, which is modified so that it produces distograms instead of atomic coordinates (Appendix A.1). The approach described here can use any oracle model that produces distogram-like outputs as part of predicting protein structure, such as AlphaFold. Although less performant than AlphaFold, we intentionally use DMPFold2 as it can much more rapidly produce predictions for batches of sequences and, importantly, doing so allows AlphaFold to be used as a highly accurate judge of designed sequences without having been directly optimized against as part of the design process. In this role, we call AlphaFold an evaluation model, $p_\alpha(y_d|x_d)$. Although AlphaFold produces a point estimate of the atomic structure, it also predicts a distogram based on amino acid β-Carbon distances.

3.5 Building the DARK framework for design

Algorithm 1 Design in Areas of Restricted Knowledge (DARK)

Input: Oracle model $p_\beta(y_d|x_d)$
Input: Iterations N
Input: Sample sizes at each iteration $m_{0:N} = \{m_0, ..., m_N\}$
1: $x_{s^0} \leftarrow$ Generate m_0 seed samples from $p_\beta(y_d|x_d)$
2: **for** $n = 1, 2, ..., N$ **do**
3: $\quad p_{\gamma^n}(x_s) \leftarrow p_{\gamma^n}(x_s|x_{s^{n-1}})$, Learn a new sequence model
4: $\quad x_{s^*} \leftarrow$ Ancestral sampling of $p_{\gamma^n}(x_s)$ $\quad \triangleright\ m_n$ new samples
5: $\quad x_{s^*} \leftarrow$ Rapid optimization of x_{s^*} IG-score using the oracle
6: $\quad x_{s^n} \leftarrow [x_{s^*}, x_{s^{n-1}}]$ Combine optimized samples with existing samples
7: **end for**

DARK (Algorithm 1) tackles the primary limitation on effectively learning models, which is having suitable sequences to train and test against. The first step is to take some initial set of samples x_{s^0} to train an initial sampling model $p_{\gamma^1}(x_{s^0})$ with parameters $\gamma^1 \in \Gamma$. We call the first set of samples the *seed samples*. Although these could be random sequences, we instead sample a small number of sequences using the simulated annealing approach from trDesign (Anishchenko et al., 2020). This kick-starts DARK by providing a small set of high-quality training examples. It also facilitates setting aside a number of the seed samples to be used as validation and test sets for evaluating the sampling models.

The sampling models are all generative sequence models that can be sampled. After training $p_{\gamma^1}(x_{s^0})$, it is used to generate a large set of samples x_{s*}. Instead of then training a new sampling model on x_{s*}, each sequence in x_{s*} is quickly optimized to improve the IG-score. This is done by a greedy hill-climb on the sequence for a set number of steps (Line 5 in Algorithm 1), which we refer to as *refinement*. In each step, a random position is mutated. If it improves the IG-score then it is kept, otherwise it is discarded. We use 3000 steps, as this was found to improve the average IG-score of x_{s*} to approximately equal to the IG-score of x_{s^0}. For simplicity, this is fixed for all steps. The result of optimizing x_{s*} is then combined with x_{s^0} to make x_{s^1}. In the second iteration, x_{s^1} is used to train a new sampling model $p_{\gamma^2}(x_{s^1})$. After some number of iterations N, the final sampling model is viewed as a strong prior over x_d, also making it a powerful generative model that can be used for *de novo* design tasks. We note that the iterative aspect of DARK share similarities to

[3]In practice, for a given length, the background model distograms can be calculated once, taking an average over 100 predictions, and then saved to be used for all IG-score calculations.

methods like Estimation of Distribution Algorithms, a common approach in model based optimization (Bengoetxea et al., 2001; Brookes et al., 2020). A method like trDesign slowly samples from some $p_(y_d|x_d)$ with a weak sequence prior. Conversely, DARK distills information from the structure predictor into the generative model in the form of high-quality sequence samples, progressively building it into a strong prior, which can be used with evaluation models like AlphaFold, enabling rapid and efficient joint sampling of high-likelihood sequences and predicted structures.

Generating seed samples and refined samples We generate $15,000$ seed examples by sampling from $p_\beta(y_d|x_d)$ using the previously described simulated annealing procedure. We sample additional examples to be used as validation and test sets, each comprising 950 samples. We use the validation set for hyperparameter optimization of the sampling models and the test model for evaluating the final chosen models at each iteration (Appendix B.1). For the second iteration we sample and refine 85,000 sequences to bring the combined total to 100,000 sequences for that iteration's the training set. For the third iteration we sample and refine 400,000 to bring the combined total to 500,000 sequences. The choice of sample sizes at each iteration was entirely dictated by computational resources available at the time and is arbitrary. Although convergence wasn't reached, 3 iterations were performed as we had no reason to expect 4+ iterations to contribute significantly to the reported results.

3.6 Deep generative architectures

In DARK, we use deep autoregressive language models for learning protein sequence distributions. Specifically, we use a standard Transformer decoder architecture (Vaswani et al., 2017; Radford et al., 2019) which has been used extensively to train state-of-the-art language models capable of generating high-quality synthetic sequence samples. For a given model $p_\gamma(x_s)$, the standard factorization is used:

$$p_\gamma(x) = \prod_{i=1}^{L} p_\gamma(x_i|x_{1:i-1}) \tag{3}$$

for sequence example $x \in x_s$. In the first iteration of DARK we use a small decoder model, which we refer to as *DARK*$_1$, a standard decoder model with 4 layers, 4 heads, and a feed-forward size of $H = 128$. To provide baseline comparisons, we compare it to a 1-layer LSTM with a hidden dimension $H = 128$ and a baseline unigram language models. After the first iteration, the second and third iteration models, DARK$_2$ and DARK$_3$ respectively, use a decoder with 12 layers, 12 heads, and $H = 768$. We also use the same architecture to train a model on natural sequences for comparison to DARK models. We refer to this model as UF50 as it is trained on 50M natural sequences from the cluster representatives of the UniRef50 protein sequence database (The UniProt Consortium, 2021). For all trained models we perform a small amount of hyper-parameter optimization on the validation set, which is also used for early stopping. Here, we focus on the 'vanilla' versions of these models in the interest of compute efficiency and to avoid unintentionally obfuscating the presented results.

3.7 Quantitatively measuring sample quality

We define high-quality samples as those that show high diversity in their sequences and predicted structures, and that have a structure predicted with high confidence by AlphaFold. For the purposes of *de novo* design, sequences should also be unlike those seen before in nature. We measure these characteristics using a few different methods, as described below. When measuring sample quality for a given model, unless otherwise stated, we calculate the metrics described here on a sample of 1000 sequences drawn from the relevant generative model.

Sample Diversity To measure the diversity of sequences and predicted structures, we use the estimated number of clusters in samples of sequences or structures. We cluster sequences using the MMseqs2 package (Steinegger & Söding, 2017) using a 30% sequence identity threshold. To cluster structures, we use AlphaFold to generate structures for each sample, and then cluster these using single-linkage hierarchical clustering, using TM-score (Zhang & Skolnick, 2005) as a dissimilarity metric.

Sample Structure Quality We define high-quality structures as those predicted with high confidence and which, like real structures, exhibit high regularity and *order*. The IG-score used in DARK provides a measure of these properties, but it is used as part of the optimization procedures. Therefore, we use the primary confidence measure produced by AlphaFold when predicting structures, the *pLDDT* score, which ranges from 0 to 100. pLDDT provides a measure of certainty and it was found that low scores (pLDDT<50) strongly suggest disorder (Tunyasuvunakool et al., 2021). We use the proportion of samples with a pLDDT>70 as a measure of high quality for the predicted structures, which we refer to as *Good+ pLDDT*, indicating "good or better" quality according to criteria in Tunyasuvunakool et al. (2021). Using AlphaFold also provides a check to ensure that the model has not learned to produce adversarial examples against the original DMPFold2 oracle.

4 RESULTS

Here, we evaluate our approach in three key ways. First, we benchmark the performance of the unconditional language models trained with DARK by likelihood. Next, we assess the quality of samples from DARK models. Finally, by way of demonstration, show that DARK combined with AlphaFold facilitates *de novo* design.

4.1 COMPARING UNCONDITIONAL LANGUAGE MODELS

Table 1: **Perplexities (PPLs) and IG-scores for unconditional protein language models.** Lower PPL is better and higher IG-score is better. IG-scores are calculated from 1000 sequence samples. IG-scores for refined (Ref) samples are included for DARK models.

Language Model	**PPL Test** $\downarrow$	**IG-score** $\uparrow$	**IG-score (Ref)** $\uparrow$
Unigram	17.69	0.27	-
Pfam HMM profiles*	11.64*	-	-
UF50	12.11	0.21	-
LSTM	9.94	0.92	-
DARK_1	6.33	1.38	1.77
DARK_2	5.73	1.53	1.81
DARK_3	**5.20**	**1.63**	**1.84**
Seed Samples (15K)	-	1.76	1.76

*Values reproduced from Ingraham et al. (2019); a conditional model conditioned on a specific sequence position of a specific family.

Baseline perplexities We show two baselines for language modelling by per-residue perplexity (Table 1). First, we include a Unigram frequency model of the seed samples with a perplexity of 17.7. We also show the average perplexity for first-order protein sequence profiles from the Pfam database (Mistry et al., 2021) calculated by Ingraham et al. (2019). Unlike all other approaches in

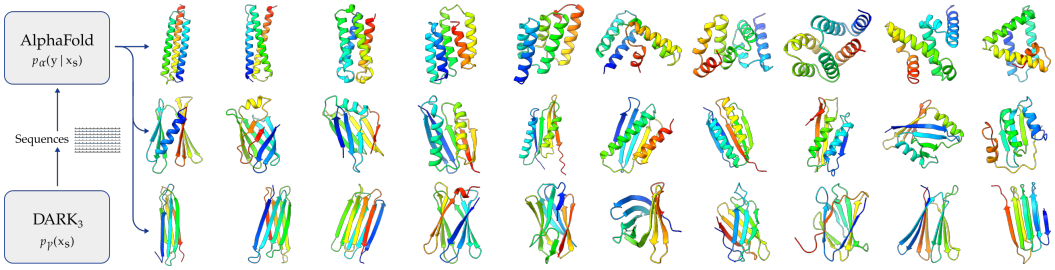

Figure 2: **DARK_3 paired with AlphaFold unconditionally samples diverse predicted structures.** We show varied examples of protein structures, predicted by AlphaFold, for sequences sampled directly from DARK_3 without any refinement. These structures have a variety of different folds, and are visualized with ChimeraX (Pettersen et al., 2021).

Table 1, it is conditional, namely, on a given protein evolutionary family and position in the family. Although this includes Cysteine where we do not, we include it as it shows how a perplexity of 11.6 can be achieved with even a simple first order model in the conditional setting.

DARK iterations improve performance All three DARK models significantly exceed the baseline performances and the performance of the UF50 model. By sampling and growing the dataset the perplexity of DARK_1 of 6.3 improves to 5.2 with the larger dataset (Table 2). This demonstrates the effectiveness of DARK and provides a strong confirmation that the core problem is one of data availability; highly performant models need a sufficient number of training sequences with with a strong signal for structure. The difference between the LSTM model's perplexity of 9.9 and DARK_1's 6.3 further confirms that architectures using self-attention are very effective for protein sequence problems. By iteration 3, the autoregressively sampled sequences have almost the same IG-score as the seed samples, and the refined samples exceed it. This strongly suggests that DARK has learned a distribution of sequences that incorporates some learned and latent structural information.

Natural sequences lead to poor performance The UF50 model provides a direct comparison to having trained on natural sequences instead of synthetic sequences, and is very similar to models like ProGEN if it was trained unconditionally (Madani et al., 2020). As expected, it performs very poorly by both perplexity and IG-score. It has a perplexity of 12.1 which is only better than the weak Unigram baseline (Table 2). Although the test set comprises synthetic sequences, we would still expect UF50 to assign high likelihoods to the test set. The synthetic test set sequences are the product of optimizing with the oracle, which is trained on real sequence-structure pairs. If UF50 learned any information regarding structure from the 50M sequences in its training set, it should also place a high likelihood on synthetic sequences, however it achieves an IG-score lower than the Unigram model suggesting that it is unlikely.

4.2 Comparing sample quality of language models using AlphaFold

High structural quality We find that DARK_3 produces samples with high structural quality (Table 2). Without any refining or conditioning, 32.3% of DARK_3 samples have Good+ pLDDT scores. Refining further increases this to 37.3% (See Appendix Figure 4 for score distributions). We compare to test set samples from two recent approaches that use a weak conditioning on structure, and find that DARK models significantly outperform them. These are not ideal comparisons, as the rival models are conditional with simpler architectures, but they show that even without conditioning, DARK achieves a high performance. We also find that DARK improves Good+ pLDDT with every iteration (Table 3). The changes are small, suggesting that a large number of iterations would be needed to match the refined score of DARK_3. For further context, we calculated the mean pLDDT scores of recently released *de novo* designed sequences out of AlphaFold's training set and found it to be high at 88.1 ($\sigma = 4.41$). This suggests that Good+ pLDDT is a good measure of a model's ability to produce samples with a quality sufficient for design tasks.

Table 2: **DARK_3 produces samples with confidently predicted structures by AlphaFold.** Good+ pLDDT scores from AlphaFold (higher is better) are shown for unconditional language models and two recent conditional models that use non-specific (weak) structure information for conditioning.

Language Model	Good+ pLDDT (%) ↑	Conditioning	Training Data
Random	1.8	-	-
gcWGAN (Karimi et al., 2020)	2.8	Structure	Natural
gramVAE (Greener et al., 2018)	21.0	Structure	Natural
UF50	15.9	-	Natural
DARK_3 *(ours)*	**32.3**	-	Synthetic
DARK_3 (Ref) *(ours)*	**37.3**	-	Synthetic

Diverse samples We find that samples from DARK models have diverse sequences and structures (Table 3). Across all iterations, clustering results suggest that only a small number of samples have similar sequences and structures.

Table 3: **Samples from DARK models have diverse sequences and predicted structures.** Sequence and structure diversity is shown with the number of clusters (of 1000 samples) estimated by sequence clustering and structure clustering respectively. Change in the Good+ pLDDT between iterations is also included.

DARK Model	Good+ pLDDT (%) ↑	Sequence Clusters ↑	Structure Clusters ↑
DARK_1	28.0	1000	829
DARK_2	31.0	973	835
DARK_3	32.3	962	822

4.3 DARK DE NOVO DESIGN

Dark samples novel sequence We find that DARK generates novel sequences w.r.t. to those in nature. We performed sequence searching on the 1000 previously analyzed samples from DARK_3 and found no strong similarity to any natural proteins in UniProt. This suggests DARK models produce samples suitable for *de novo* design tasks.

Designing in an unconditional way We provide a direct but small example of how DARK can be used for developing *de novo* design approaches. The 4-helix bundle is a common protein fold (topology) seen in design studies and in nature (Orengo et al., 1997). We search for and find a 4-helix bundle in the sampled sequences from DARK_3 with a high pLDDT (91.4), similar to the pLDDT of existing *de novo* proteins (see previous section). We compare the predicted structure to the structure predicted by the Rosetta AbinitoRelax method. We find that AbinitoRelax folds the sequence into very nearly the same structure as AlphaFold. The best predicted structure of 10,000 structure predictions (decoys) by Rosetta all-atom score (-279 Rosetta Energy Units) has a low root-mean-square deviation of 3.11Å (by α-carbons) compared to AlphaFold's prediction. Both predicted structures and the spread of decoy energy values are visualized in Appendix Figure 3 & 5 respectively. This suggests a correct design, suggesting AlphaFold and DARK can be used to build future, more complex, design approaches. Clearly, searching a set of samples is not a robust way of specifying a desired protein structure. However, given the success of DARK without conditioning, it clearly shows that it can be used to develop further generative approaches to *de novo* design that neurally sample sequence and structure jointly. Controlled generation could be explored using learned latent variables (Chen et al., 2016) or with weak conditioning information derived from DARK samples.

5 CONCLUSION

In this work, we show that it is possible for deep generative models to learn a distribution of protein sequences that incorporates learned relationships with structure without any conditioning information, resulting in samples with confidently predicted structures. We achieve this by leveraging the learned sequence-structure information in state-of-the-art structure predictors to produce synthetic samples for training. Jointly sampling a sequence from DARK_3 and predicting its structure with AlphaFold, excluding relaxation, demonstrates a completely neural and unconditional way of sampling a protein from the joint sequence-structure space $p(x, y)$. This provides a path for future generative models that tackle *de novo* design tasks by exploring topics like controlled generation with DARK.

Although further iterations of the DARK framework improved the predicted structures of samples, they did so slowly, at least when using AlphaFold as a judge of quality. Identifying better ways to generate and learn structure information from synthetic sequences remains an important research direction. Furthermore, across the DARK framework there are a large number of hyper-parameters. Future work would benefit from automatic approaches to setting these and, more generally, an exploration of their benefits and limitations.

ACKNOWLEDGMENTS

Left blank for anonymity during double-blind peer review.

REPRODUCIBILITY STATEMENT

In order to guarantee reproduciblity of this work, we provide a detailed appendix with specific implementation and methodological details which is sufficient to reproduce the results in this work. Following release of the paper, we will release both a code repository containing the source code for the DARK models as well as the associated scoring functions and analysis scripts. The training data and model parameters will be made available from our servers due to GitHub file size limits.

REFERENCES

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

Namrata Anand and Po-Ssu Huang. Generative modeling for protein structures. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7505–7516, 2018.

Antonina Andreeva, Eugene Kulesha, Julian Gough, and Alexey G Murzin. The scop database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic acids research*, 48(D1):D376–D382, 2020.

Ivan Anishchenko, Tamuka Martin Chidyausiku, Sergey Ovchinnikov, Samuel J Pellock, and David Baker. De novo protein design by deep network hallucination. *bioRxiv*, 2020.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Endika Bengoetxea, Pedro Larranaga, Isabelle Bloch, and Aymeric Perchant. Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 454–469. Springer, 2001.

Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

David Brookes, Akosua Busia, Clara Fannjiang, Kevin Murphy, and Jennifer Listgarten. A view of estimation of distribution algorithms through the lens of expectation-maximization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 189–190, 2020.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, et al. Rcsb protein data bank: powerful new tools for exploring 3d structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic acids research*, 49(D1):D437–D451, 2021.

Yue Cao, Payel Das, Vijil Chenthamarakshan, Pin-Yu Chen, Igor Melnyk, and Yang Shen. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *International Conference on Machine Learning*, pp. 1261–1271. PMLR, 2021.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2180–2188, 2016.

Aaron Chevalier, Daniel-Adriano Silva, Gabriel J Rocklin, Derrick R Hicks, Renan Vergara, Patience Murapa, Steffen M Bernard, Lu Zhang, Kwok-Ho Lam, Guorui Yao, et al. Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674):74–79, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.

Joe G Greener, Lewis Moffat, and David T Jones. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, 8(1):1–12, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in Neural Information Processing Systems*, 32:15820–15831, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Shaun M Kandathil, Joe G Greener, Andy M Lau, and David T Jones. Deep learning-based prediction of protein structure using learned representations of multiple sequence alignments (version 1). *Biorxiv*, pp. 2020–11, 2020.

Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. *Journal of Chemical Information and Modeling*, 60(12):5667–5681, 2020.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing dna with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

Neil P King, William Sheffler, Michael R Sawaya, Breanna S Vollmar, John P Sumida, Ingemar André, Tamir Gonen, Todd O Yeates, and David Baker. Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science*, 336(6085):1171–1174, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ivan V Korendovych and William F DeGrado. De novo protein design, a retrospective. *Quarterly reviews of biophysics*, 53, 2020.

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

Johannes Linder, Nicholas Bogard, Alexander B Rosenberg, and Georg Seelig. A generative neural network for maximizing fitness and diversity of synthetic dna and protein sequences. *Cell Systems*, 11(1):49–62, 2020.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Enrique Marcos, Tamuka M Chidyausiku, Andrew C McShan, Thomas Evangelidis, Santrupti Nerli, Lauren Carter, Lucas G Nivón, Audrey Davis, Gustav Oberdorfer, Konstantinos Tripsianes, et al. De novo design of a non-local β-sheet protein with high stability and accuracy. *Nature structural & molecular biology*, 25(11):1028–1034, 2018.

Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, 2021.

Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, David Baker, and Sergey Ovchinnikov. Protein sequence design by conformational landscape optimization. *Proceedings of the National Academy of Sciences*, 118 (11), 2021.

Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Nelson Perdigão, Julian Heinrich, Christian Stolte, Kenneth S Sabir, Michael J Buckley, Bruce Tabor, Beth Signal, Brian S Gloss, Christopher J Hammang, Burkhard Rost, et al. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences*, 112(52):15898–15903, 2015.

Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Elaine C Meng, Gregory S Couch, Tristan I Croll, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Structure visualization for researchers, educators, and developers. *Protein Science*, 30(1):70–82, 2021.

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

Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. *bioRxiv*, 2021.

Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Sari Sabban and Mikhail Markovsky. Ramanet: Computational de novo helical protein backbone design using a long short-term memory generative neural network. *F1000Research*, 9(298):298, 2020.

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):1–11, 2021.

Daniel-Adriano Silva, Shawn Yu, Umut Y Ulge, Jamie B Spangler, Kevin M Jude, Carlos Labão-Almeida, Lestat R Ali, Alfredo Quijano-Rubio, Mikel Ruterbusch, Isabel Leung, et al. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7738):186–191, 2019.

Sam Sinai and Eric D Kelsic. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv preprint arXiv:2010.10614*, 2020.

Jedediah M Singer, Scott Novotney, Devin Strickland, Hugh K Haddox, Nicholas Leiby, Gabriel J Rocklin, Cameron M Chow, Anindya Roy, Asim K Bera, Francis C Motta, et al. Large-scale design and refinement of stable proteins using sequence-only models. *bioRxiv*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible design of novel proteins using graph neural networks. *BioRxiv*, pp. 868935, 2020.

William R Taylor, Vijayalakshmi Chelliah, Siv Midtun Hollup, James T MacDonald, and Inge Jonassen. Probing the "dark matter" of protein fold space. *Structure*, 17(9):1244–1252, 2009.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021.

Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, et al. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Anastassia A Vorobieva, Paul White, Binyong Liang, Jim E Horne, Asim K Bera, Cameron M Chow, Stacey Gerben, Sinduja Marx, Alex Kang, Alyssa Q Stiving, et al. De novo design of transmembrane β barrels. *Science*, 371(6531), 2021.

Fengbin Wang, Ordy Gnewou, Charles Modlin, Leticia C Beltran, Chunfu Xu, Zhangli Su, Puneet Juneja, Gevorg Grigoryan, Edward H Egelman, and Vincent P Conticello. Structural analysis of cross α-helical nanotubes provides insight into the designability of filamentous peptide nanomaterials. *Nature communications*, 12(1):1–14, 2021.

Jingxue Wang, Huali Cao, John ZH Zhang, and Yifei Qi. Computational protein design with deep learning neural networks. *Scientific reports*, 8(1):1–9, 2018.

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

Derek N Woolfson. A brief history of de novo protein design: minimal, rational, and computational. *Journal of Molecular Biology*, pp. 167160, 2021.

Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine learning in protein engineering. *arXiv preprint arXiv:1811.10775*, 2018.

Chi-Hua Yu and Markus J Buehler. Sonification based de novo protein design using artificial intelligence, structure prediction, and analysis using molecular modeling. *APL bioengineering*, 4(1): 016108, 2020.

Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

Jianfu Zhou, Alexandra E Panaitiu, and Gevorg Grigoryan. A general-purpose protein design framework based on mining sequence–structure relationships in known protein structures. *Proceedings of the National Academy of Sciences*, 117(2):1059–1068, 2020.

APPENDIX

A IMPLEMENTATION AND TRAINING DETAILS

A.1 ORACLE MODEL AND BACKGROUND MODEL

We use the DMPfold2 model for the $p_\beta(y|x)$ oracle (Kandathil et al., 2020). We received the trained model as well as the training and validation data described in Kandathil et al. (2020) via correspondence with the authors. We note that the code and pre-print for a more recent but non-probabilistic version appears to be available[4], but we refer here to the CASP14 version of DMPfold2 described in the first pre-print (Kandathil et al., 2020), which outputs distograms.

DMPfold2 was chosen as an oracle as it reports favourable performance and, importantly, its ability to perform fast inference directly from an input alignment, or in our case, a single sequence. It uses a combination of two GRUs (Cho et al., 2014) followed by a residual network to predict distograms from a multiple sequence alignment. The input is of shape $[A, L, 22]$ where the A dimension is the number of sequences in the alignment. First, a stack of unidirectional GRUs reduces A to a fixed size, resulting in an output array sized $[L, 512]$, that is fed as input to a stack of bidirectional GRUs that produces the final representation of the input alignment. This representation is then fed to a convolutional ResNet, which produces the final outputs (see below).

We refactored the code for the DMPfold2 model in order to perform batched inference on individual sequences. We find that the IG-score for a single sequence ($L = 100$) can be calculated in approximately 0.06s. For a batch of 100 sequences this reduces to 0.01s per sequence.

Distogram outputs from DMPfold2 From a single input sequence of length L, DMPfold2 produces an output tensor of size $[L, L, 104]$. The 104D third axis is split into four separate discrete distributions which are individually softmaxed, being sized $[L, L, 2]$ and three of $[L, L, 34]$. The output tensor contains predictions of residue-residue hydrogen bonding, residue-residue β-carbon ($C\beta$) distance distributions (distograms), and backbone ϕ and ψ torsion angle distributions. See Kandathil et al. (2020) for further details.

DMPfold2 Background Model To implement the IG-score we train a background network, $p_\epsilon(y_d)$, as per Anishchenko et al. (2020), using the DMPfold2 architecture and author-provided training data. This model is distinguished with the subscript ϵ. Training the background model constituted replacing the input sequences during training with 64D Gaussian noise, and altering the input layer accordingly, but otherwise following the training procedure described in the DMPfold2 paper. We pre-calculate the background values for a given sequence length by feeding noise to the network 100 times and taking the average of the predictions, leaving a final tensor of size $[L, L, 104]$ containing the resulting predictions. Although this can be calculated on the fly, for simplicity we pre-calculate and save them once. As we focus on $L = 100$, this amounts to one saved array of size $[100, 100, 104]$.

Using DMPfold2 to calculate the IG-score The IG-score is calculated by first predicting the output tensor for a sequence with DMPfold2. Next, D_{KL} is calculated between each of the four distograms in the tensor and their corresponding pre-calculated background versions. This results

[4]https://www.github.com/psipred/DMPfold2

in four $[L, L, 1]$ tensors, one for each distogram, containing the scalar D_{KL} values. For each tensor, the mean D_{KL} is taken across both dimensions, resulting in four scalars. These four D_{KL} values are added together with equal weighting. Finally, the IG-score is this value minus the D_{KL} between the unigram frequency distribution in the predicted sequence and the corresponding distribution in high resolution protein structures from the PDB. This results in a single scalar D_{KL}. The latter values are sourced from the publicly available trDesign code repository[5].

A.2 AUTOREGRESSIVE MODELS

We build the autoregressive neural network models used in this work with PyTorch (Paszke et al., 2019). We also use several utilities from the NLTK (Bird et al., 2009) package. For all neural network architectures we do mini-batch gradient descent against negative log likelihood (NLL) loss. We use the Adam optimizer (Kingma & Ba, 2014) with default β values of 0.9 & 0.999. Early stopping is performed using the validation set. A batch size of 64 is used with a learning rate of 1×10^{-3} for the RNN baseline, and 1×10^{-5} for self-attention models. The learning rate was not changed according to a schedule as it was not found to provide significant benefit, and so was left static. All training was done with automatic mixed precision as it brought a significant improvement in computational efficiency. Below we describe the architectures for the best RNN baseline model and for the self-attention models.

Inputs For all language models trained on synthetic data, the input is a batch of B sequences in an array sized $[B, L]$. This contains 22 integers corresponding to 20 amino acids, a start token, and a stop token. The length L is the length of the protein sequence with a start token placed at the beginning of the sequence. As is standard for autoregressive models, the target sequence used for calculating the loss is the protein sequence with a stop token placed at the end. All neural networks use embedding layers that use these as integer indices for embedding vectors learned during training.

RNN architecture The best RNN model uses an embedding layer, learned during training, of size 64. The embedded sequence of size $[B, L, 64]$ is fed into an LSTM model with a hidden size H of 128 and 1 layer (Hochreiter & Schmidhuber, 1997). The hidden vector output at each time step of the LSTM is then fed to a linear layer ($H = 128$) $\rightarrow$ ReLU activation $\rightarrow$ linear layer ($H = 22$). The resulting final array, sized $[B, L, 22]$, is then log-softmaxed.

Xfmr architecture The Big and Small Xmfr models use a common architecture but with different numbers of parameters. We describe the architecture using the sizes of the Big model first. An embedding layer of size 384 is concatenated with a positional encoding of size 384 (Vaswani et al., 2017), resulting in an array of size $[B, L, 784]$. The positional encoding uses a maximum length of 101. In comparison, the Small model uses an embedding size of 64, so the input to the following decoder has 128 dimensions in the final axis. We use a standard decoder architecture for autoregressive multi-headed self-attention with causal masking (Vaswani et al., 2017; Radford et al., 2019). This includes 10% Dropout (Srivastava et al., 2014) and the use of Layer Normalization (Ba et al., 2016). We use the Fast Transformers library (Katharopoulos et al., 2020) for implementing models. The Big model uses 12 layers with $H = 784$ and 12 heads, and the Small model uses 4 layers with $H = 128$ and 4 heads. For both models the output of the decoder goes into a linear layer ($H = 128$) $\rightarrow$ ReLU activation $\rightarrow$ linear layer ($H = 22$). The output is produced is also a $[B, L, 22]$ array that is log-softmaxed.

For both RNN language models and self-attention language models, additional decoding strategies beyond simple autoregressive sampling like Top-K sampling (Welleck et al., 2019) were not found to significantly improve results. We mask out cysteine when sampling from DARK models, however we include it in the sizing of the input embeddings and output size to maintain consistency with other approaches. The output array of size 22 accounts for the 20 amino acids, a start, and a stop token. Further investigation of more complex decoding schemes, as well as the use of different positional encoding methods and types of attention, are potential avenues for future research.

[5]https://github.com/gjoni/trDesign

A.3 TRAINING AN UNCONDITIONAL MODEL WITH NATURAL SEQUENCES

Here we describe further architecture and training details of the 'UF50' model used to benchmark the DARK models produced in this work. This model provides a direct comparison of training with natural sequences and training with synthetic proteins from DARK. UF50 is an unconditional language model that uses the same training procedure and architecture as the DARK_3 model, with some modifications, and trained on sequences from nature. Specifically, the model is trained on the representative sequences for each cluster in the UniRef50 database (The UniProt Consortium, 2021), yielding a training set of 50,106,395 sequences. UF50 provides a benchmark for an approach similar to the ProGEN model (Madani et al., 2020), if ProGEN was an unconditional model. ProGEN is conditioned on a variety of evolutionary and functional information for the task of protein engineering, making it not directly comparable to the work performed here.

We now describe the modifications used in the UF50 model. During training, missing amino acids and non-canonical amino acids are mapped to an additional 'UNK' token which does not contribute to the loss. Thus, the final vocabulary size is 24. Batches are dynamically padded to the length of the longest sequence, and padded positions are masked for the loss calculation. To reduce the GPU memory required for long sequences, any sequences longer than 100 amino acids (start/stop not included) are clipped to 100. For the target sequence, instead of placing a stop token at position 101, the amino acid sequence in position 101 from the original non-clipped sequence is used. In this way, due to the causal masking, this is exactly the same information available to the model as if it was not using clipped sequences, within the range of 101 amino acids. As all evaluated sequences are of length 100, the clipping has no effect on any conclusions drawn. To provide a level playing field, UF50 uses the same validation set as DARK models. We also use rejection sampling to sample 1000 sequences of length 100 for evaluation and comparison. Additional decoding strategies like Top-K sampling and placing a temperature factor on the next-token softmax were explored but not found to aid performance.

B SEQUENCE AND STRUCTURE ANALYSIS

B.1 VALIDATING THE VALIDATION AND TEST SET

Ideally each seed sample from the black-box oracle could be viewed as identical and independently distributed (i.i.d.). However, given the sometimes unexpected pathologies (e.g. adversarial examples) that can be present in neural networks, we directly measure the diversity of the training set sequences as well as look for any direct overlap of the validation and test sets with the training set. As can be seen in Table 4, the training set sequences begin to form clusters at around 30% sequence identity.

As a method of comparing generative models between architectures and iterations of DARK, a test set and a validation set of 950 samples each are generated, using the simulated annealing approach of trDesign. We check that the two sets do not share extremely similar or identical sequence samples with the seed set, producing artificially high performance metrics. As such, validation and test set sequences are separately compared to all 15,000 sequences in the seed sample set. This is done using the MMseqs2 (Steinegger & Söding, 2017) search tool with a sensitivity parameter of 7 and one iteration. Given the clustering results of the seed samples (Table 4), we expect a high proportion of the sequences to share some small similarity to sequences in the seed sample set. This was found to be the case; from the validation and test set, 854 out of the 950 sequences in each set were found to have some similarity to sequences in the training set. The mean sequence identity between the similar sequences for both sets was low, being 31% for both. The highest sequence identity found in the validation set was 68%, and for the test set 60%. These results are as would be expected; they suggest that both sets are sampled from the same distribution as the seed set while not containing any near-identical copies. This is acceptable, given that this is an unsupervised task using synthetic samples, where the aim is to learn that distribution. We make the distinction as had natural sequences been used a more stringent test and validation splitting based on structure and evolutionary relationships would be required.

As part of investigating the effects of DARK iterations, we also performed this search process with the test set for the 500,000 examples of the third iteration's training set. The number of examples in the test set that had any detected hits increased a small amount, from 854 ($N = 1$) to 922. The

mean sequence identity of the hits increased slightly from 31% to 33% and the highest identity hit increased from 60% to 66%. Again, these values are acceptable, and even potentially a positive sign. One likely cause for the increase in the performance of models across iterations of DARK is that the area of sequence space covered by the training set is progressively growing in an unsupervised manner. Even from a pessimistic perspective, if the $N = 3$ model has better likelihood estimates on the test set because the previous iterations generated training samples closer in space to the test set, then it suggests that the iterations are indeed expanding and filling in unseen parts of the distribution.

B.2 Predicting structure with AlphaFold

We use the publicly available version of AlphaFold[6] for predicting structures and as part of a quality metric. In both cases, we use the standard AlphaFold inference pipeline, meaning we predict with the five 'CASP14' models, take the predicted structure with the highest pLDDT as the final result, and perform relaxation. We use adapted inference scripts to accommodate for predicting with only a single sequence and without using the template pipeline or the multiple sequence alignment (MSA) pipeline. This consisted of setting all models to use the same default configuration, as they all use variants on the default, and then setting all to not use templates. The MSA clusters are reduced to 1 sequence, being the single input sequence, and the deterministic flag is set to true. The number of recycles is set to 4. Following structure prediction, the standard relaxation is performed for the best final structure. We note that relaxation is not necessary to produce a final structure, thus it can be said that a trained language model, like those we present, with AlphaFold provides a direct way to sample from a joint distribution of sequence and structure, and without additional conditioning information. Predicting structure with AlphaFold provides the pLDDT scores at each residue position. We take the average of these for the pLDDT score of a given sequence. Structures are visualized with ChimeraX (Pettersen et al., 2021).

B.3 Structural quality of previously designed de novo sequences

As we use the pLDDT from AlphaFold as a metric for measuring structure quality, we wish to double-check that AlphaFold assigns good or better pLDDT scores to *de novo* designed sequences in the PDB that were not used as a part of its training set. We manually curate a list of 26 of these sequences from the PDB. The PDB IDs containing these are 6MRR, 6MRS, 6NUK, 6MSP, 6XWI, 6XXV, 6VTW, 7KBQ, 6D0T, 7JH6, 6XEH, 6X9Z, 6WVS, 6E5C, 6DS9, 5TX8, 5W9F, 6YWD, 6Z35, 6W6X, 6W90, 6WI5, 6W3W, 6VGB, 6W70, 6YQY.

This is not a comprehensive list as, amongst other reasons, some sequences were not included if they were very sequence-similar to an already selected structure or if they are a large tandem repeat structure (e.g. 'armadillo repeats') to avoid biasing the average. This is still a biased set of structures as they are the product of whichever goal a given study was investigating, but the pLDDT scores are still informative. We note that although the PDB has a 'DE NOVO PROTEIN' classification, this includes engineered, designed, and *de novo* designed proteins. The average calculated pLDDT of all sequences in the list is 88.1 ($\sigma = 4.41$), a good score, as is expected.

B.4 Benchmarking weakly conditioned methods

We provide results for benchmarking with Good+ pLDDT proportions on two recent weakly conditioned generative protein design models, which we refer to as gramVAE (Greener et al., 2018) and gcWGAN (Karimi et al., 2020). These both use a fold level description that we call weak as they are far removed from the explicit structural information provided in fixed backbone models (Ingraham et al., 2019; Strokach et al., 2020; Cao et al., 2021). We stress that these models are not directly comparable to the unconditional models explored in this work. Nonetheless, we provide these benchmarks as they provide context and directly show that the DARK models have incorporated a latent understanding of structure into their learned sequence distributions, as they produce sequences with confidently predicted structures to a significantly greater degree than either weakly conditioned model.

[6]https://github.com/deepmind/alphafold

The gramVAE[7] uses a context-free grammar of SCOP protein folds (Andreeva et al., 2020) as a conditioning variable for a Variational AutoEncoder (VAE). Both encoder and decoder use non-linear layers and sequences are padded to a maximum length of 140. The gcWGAN is a conditional Wasserstein Generative Adversarial Network that also pads length to use fixed-sized networks (Karimi et al., 2020). It conditions on a small latent representation of folds, learned from performing kernel PCA on an all-vs-all distance matrix between all folds in the dataset, where the kernel used is symmetric TM-score (Zhang & Skolnick, 2005) between two folds. The publicly available implementation[8] provides 600 sequences generated for the 6 folds (100 sequence each) in its test set using the gcWGAN and the gramVAE in a head to head comparison. Being publicly available and a direct comparison, we use these provided sequences to calculate Good+ pLDDT proportions for both models. The gramVAE achieves 21.0% and the gcWGAN achieves 2.83%.

We note that the majority of the folds in the gcWGAN test set are in the gramVAE training set, which likely explains part of the performance difference. We are not concerned by this as the DARK models still outperform the gramVAE with an artificial boost. We also note that the gcWGAN model does not discretize the output of the generator and use any approaches to backpropagating through the discrete output variable (Bengio et al., 2013; Jang et al., 2016). This provides the discriminator a trivial learning scheme: place the decision boundaries at the edge of the probability simplex of tokens at any position. This may explain part of the gcWGAN's poor performance. In both cases, we recognize that these models are not ideally suited for sequence problems and have reported poor performance elsewhere (Cao et al., 2021). We expect future weak conditioning methods will provide more competitive benchmarks. Ideally, we hope that future investigations into novel unconditional models will provide direct benchmarks, establishing the best approach to unconditional modelling.

B.5 DARK MODEL SAMPLES ARE DISTINCT FROM NATURAL PROTEINS

We find that samples from the DARK_3 model are distinct from sequences in nature. A sequence search against UniRef100 of the 1000 evaluated sequence samples (without refinement) returned no hits for all sequences except one, which had 5 weak hits. Two pairs of these 5 sequences were extremely sequence-similar. Repeating the sequence search with a different tool, HHblits (Remmert et al., 2012), no natural sequences were found that met the HHblits default threshold value (an E-value below 1×10^{-3}).

B.6 SEQUENCE SEARCHING AND CLUSTERING

We use the MMseqs2 package (Steinegger & Söding, 2017) to perform sequence searches and sequence clustering. For clustering, the default clustering pipeline and settings are used with sequence identity as the threshold. If not specified, this is assumed to be a threshold of 30% sequence identity. For sequence searching we use the default pipeline and a high sensitivity parameter of 7.0 with 1 iteration against UniRef100 (03-2021 release) (The UniProt Consortium, 2021). Any sequence searching done with HHblits (Remmert et al., 2012) uses an E-value cut-off of 1×10^{-3} with 1 iteration against UniRef30 (06-2021 release) (Mirdita et al., 2017).

B.7 STRUCTURE CLUSTERING

To cluster predicted protein structures we perform single-linkage agglomerative clustering using TM-score (Zhang & Skolnick, 2005) as a similarity metric, meaning an all-against-all matrix of TM-scores is calculated for each structure in the set of 1000 samples. We use a TM-score of 0.4 as the cutoff for selecting best cluster centroids and a TM-score of 0.65 for single-linkage clustering around the centroids.

C ADDITIONAL FIGURES AND TABLES

[7] https://github.com/psipred/protein-vae

[8] https://github.com/Shen-Lab/gcWGAN

Table 4: **Diversity of the sequence examples in the training set.** Diversity at each iteration is measured by number of sequence clusters found by clustering with four different sequence identity (Seq. ID) thresholds. The number of training examples (Train Ex.) at each iteration is included.

Iteration	99% Seq. ID	90% Seq. ID	50% Seq. ID	30% Seq. ID	Train Ex.
$N = 1$	15,000	15,000	14,994	11,076	15,000
$N = 2$	100,000	100,000	99,942	66,509	100,000
$N = 3$	500,000	500,000	498,077	231,425	500,000

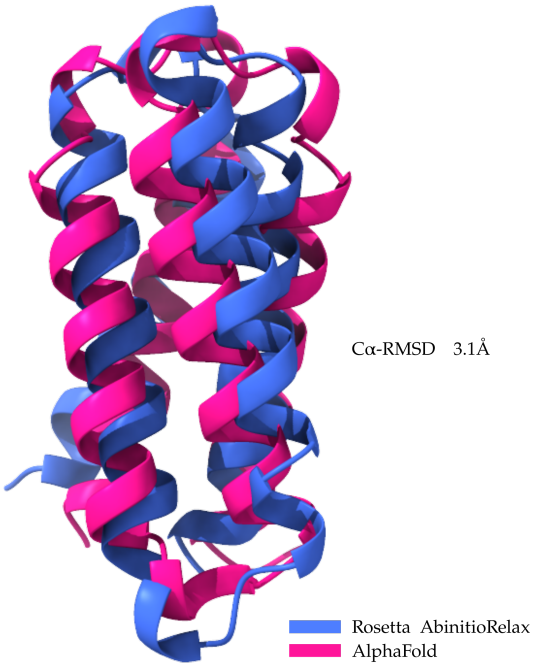

Figure 3: **Agreement between Rosetta and AlphaFold structure prediction.** Predicted structures of a sequence sample (ID 209a) using AlphaFold and Rosetta AbinitioRelax. Both approaches are in agreement, producing close to the same structure. For the latter, the best predicted structure with the lowest Rosetta all-atom score is used, being -279. The root-mean-square deviation between α carbons in the two different predictions is included.

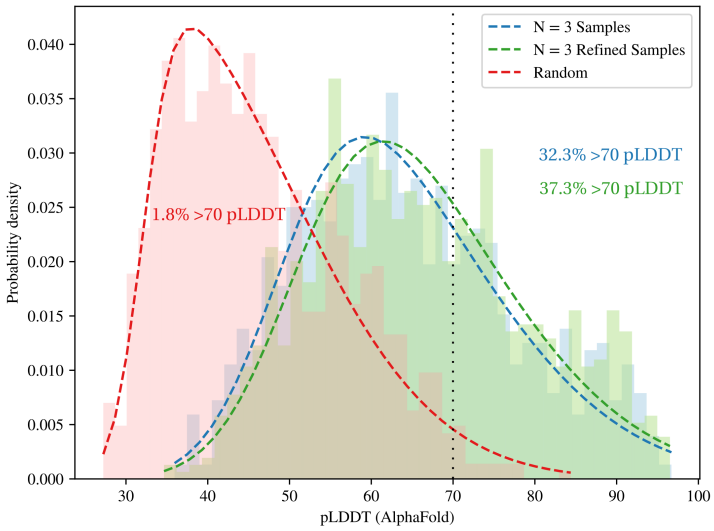

Figure 4: **DARK models generate samples with confidently predicted structures.** A Normalized histograms of the pLDDT scores produced by AlphaFold when predicting structures for three sets of sequence samples, each containing 1000 sequences. A skew-normal is fit to each set and plotted for clarity. *Random* refers to uniformly random sequences of $L = 100$. The Good+ pLDDT (pLDDT$>$ 70) proportion is also included. $N = 3$ refers to the DARK$_3$ model.

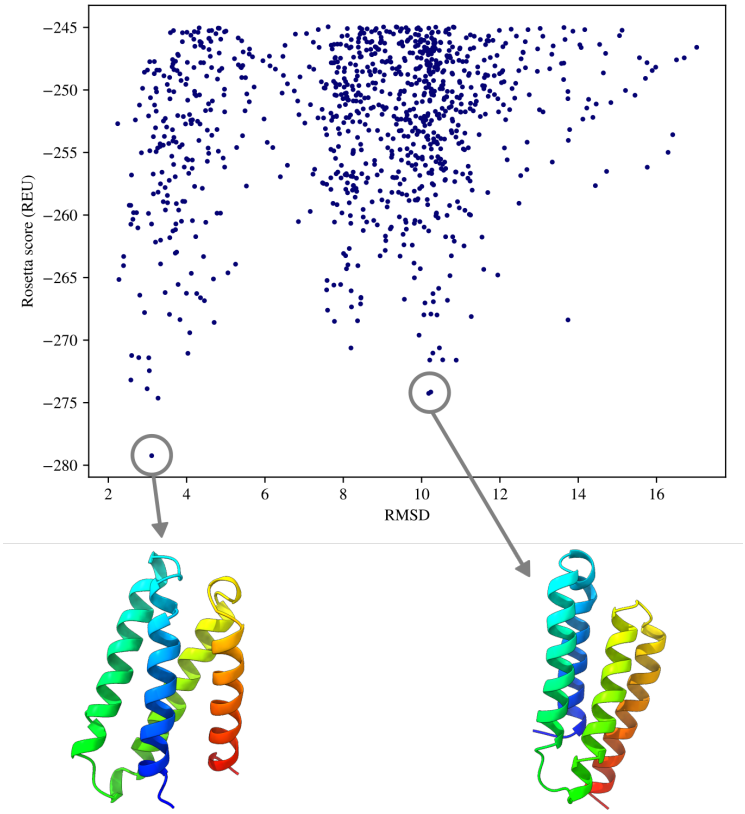

Figure 5: **Rosetta predicts correct fold and potential alternative conformation.** The top 1000 predicted structures (by Rosetta score) for a sampled sequence, of 10000 structures, produced by the Rosetta AbinitioRelax protocol. Rosetta scores are expressed in Rosetta Energy Units (REU; lower values are better). RMSD is root-mean-square deviation (RMSD) in Å from the AlphaFold structure prediction of the sampled sequence using α-carbon atoms. The best model is bottom left but a second peak suggests an alternative conformation may be possible (bottom right). We note a similarity to the multi-peak distributions presented in Figures 2B & 2C of Norn et al. (2021).