MULTI-SEGMENT PRESERVING SAMPLING FOR DEEP MANIFOLD SAMPLER

Daniel Berenberg[1,2], **Jae Hyeon Lee**[1], **Simon Kelow**[1], **Ji Won Park**[1], **Andrew Watkins**[1], **Vladimir Gligorijević**[1], **Richard Bonneau**[1], **Stephen Ra**[1], and **Kyunghyun Cho**[1,2,3,4]

[1]Prescient Design, Genentech
[2]Department of Computer Science, Courant Institute of Mathematical Sciences, New York University
[3]Center for Data Science, New York University
[4]CIFAR Fellow
berenberg.daniel@gene.com

ABSTRACT

Deep generative modeling for biological sequences presents a unique challenge in reconciling the bias-variance trade-off between explicit biological insight and model flexibility. The deep manifold sampler (Gligorijević et al., 2021) was recently proposed as a means to iteratively sample variable-length protein sequences. Sampling was done by exploiting the gradients from a function predictor trained on top of the manifold sampler. In this work, we introduce an alternative approach to guided sampling that enables the direct inclusion of domain-specific knowledge by designating preserved and non-preserved segments along the input sequence, thereby restricting variation to only select regions. We call this method "multi-segment preserving sampling" and present its effectiveness in the context of antibody design. We train two models: a deep manifold sampler and a GPT-2 language model on nearly six million heavy chain sequences annotated with the *IGHV1-18* gene. During sampling, we restrict variation to only the complementarity-determining region 3 (CDR3) of the input. We obtain log probability scores from a GPT-2 model for each sampled CDR3 and demonstrate that multi-segment preserving sampling generates reasonable designs while maintaining the desired, preserved regions.

1 INTRODUCTION

Protein sequence families, particularly antibodies, have both well-conserved and variable regions. In antibodies, the heavy and light chain sequences consist of highly conserved regions known as the framework as well as an array of distinct hypervariable loops, known as complementarity-determining regions (CDRs) (Reczko et al., 1995). Despite the intrinsic variability of CDRs, conditional variation is often conferred by the gene locus admitting the protein (Kelow et al., 2020). Much of an antibody's antigen-binding affinity is owed to the CDRs, while the framework remains fixed or requires minimal change (Kuroda et al., 2012). For *in silico* modeling, integrating these established aspects of structure and binding can drive the development of better *in situ* antibody therapeutic design (Chiu et al., 2019). While work in protein language modeling suggests that models can learn these evolutionary conservation rules (Rives et al., 2021; Elnaggar et al., 2021; Madani et al., 2020), it is an open challenge as to how to explicitly incorporate prior insight at test-time generation, such as sequence-level annotations (Dunbar and Deane, 2015), to restrict sampling in certain segments.

The deep manifold sampler was recently proposed as an effective method to sample novel sequences by iterative, optionally gradient-guided steps, of sequence denoising (Gligorijević et al., 2021). Empirically, gradient-based guided sampling was shown to selectively encourage changes in functional sites, implicitly leaving non-functional regions unperturbed. In this work, we propose an alternative to the gradient-based guided design procedure in which predefined regions of a sequence are explicitly preserved, leaving sampling to take place in *a priori* known notable sequence regions. We

conduct an experiment on antibody sequences to demonstrate the deep manifold sampler's ability to focus sampling on a subset of sequence positions. We do so by deliberately corrupting select regions of antibody sequences, that correspond to CDRs, and evaluating the length distribution and composition of sampled CDRs.

2 BACKGROUND: THE DEEP MANIFOLD SAMPLER

The deep manifold sampler (Gligorijević et al., 2021) is a denoising autoencoder (DAE) specialized to handle variable-length sequences. As with a typical DAE (Vincent et al., 2010), the deep manifold sampler consists of three modules; a corruption process $C(\tilde{x}|x)$, an encoder F and a decoder G. Unlike the usual DAE however, the deep manifold sampler has an extra module that determines the change in the length, to which we refer as the "length conversion" (Shu et al., 2020).

The deep manifold sampler assumes as input a sequence of discrete tokens, $x = (x_1, x_2, \ldots, x_L)$, where each token x_t is an item from a finite vocabulary V of unique words or subwords. In the case of protein sequence modeling, V consists of all unique amino acids. The sequence x is corrupted with the corruption process C, resulting in a *noisy* input sequence $\tilde{x} \sim C(\tilde{x}|x)$. This corruption process can be arbitrary as long as it is largely local and unstructured. It may even alter the length of the sequence, $|x| \neq |\tilde{x}|$.

The encoder F turns the corrupted sequence $\tilde{x}$ into a set of hidden vectors, $h = (h_1, h_2, \ldots, h_{|\tilde{x}|})$, where $h_t \in \mathbb{R}^d$. The encoder can be implemented using any of the widely-used deep architectures, such as transformers (Vaswani et al., 2017), convolutional networks (Gehring et al., 2017) and recurrent networks (Sutskever et al., 2014; Bahdanau et al., 2014). In this work, we follow the original deep manifold sampler's encoder, which was implemented as a transformer.

The hidden vectors are pooled to form a single-vector representation:

$$\bar{h} = \frac{1}{|\tilde{x}|} \sum_{t=1}^{|\tilde{x}|} h_t.$$

This pooled representation is used by the length conversion to predict the change in the length. At training time, this length change predictor is trained to output $\Delta l^* = |\tilde{x}| - |x|$. When we sample sequences from the deep manifold sampler after training, we use the predicted change Δl to adjust the size of the hidden vector set. The adjusted hidden vector set consists of $|\tilde{x}| + \Delta l$ hidden vectors, $z = (z_1, \ldots, z_{|\tilde{x}|+\Delta l})$, where

$$z_t = \sum_{t'=1}^{|\tilde{x}|} \omega_{t,t'} h_{t'} \tag{1}$$

with

$$w_{t,t'} \propto \frac{-1}{\exp(\sigma)} \left(t' - \frac{|\tilde{x}|}{|\tilde{x}| + \Delta l} t \right)^2 .$$

The decoder G then takes this transformed hidden vector sequence z and outputs a corresponding sequence of logit vectors, $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_{|\tilde{x}|+\Delta l})$, where $\tilde{y}_t \in \mathbb{R}^{|V|}$. These logits are turned into probability distributions over the vocabulary V in many different ways. The original deep manifold sampler implements a non-autoregressive approach (Gu et al., 2017; Lee et al., 2018), where each logit is independently turned into a distribution:

$$p(y_t = v|\tilde{x}, \Delta l) = \frac{\exp\left(\tilde{y}_t^v + b^v\right)}{\sum_{v' \in V} \exp\left(\tilde{y}_t^{v'} + b^{v'}\right)}, \tag{2}$$

where b^v is a bias for token v.

It is however also possible to use these logits together with a more powerful output module, such as a conditional random field (CRF; Lafferty et al. 2001), as was recently done in (Yi et al., 2021), and autoregressive language models (Mikolov et al., 2010). For experiments in this paper, we use a variant of the deep manifold sampler in which we use a CRF at the end of the decoder.

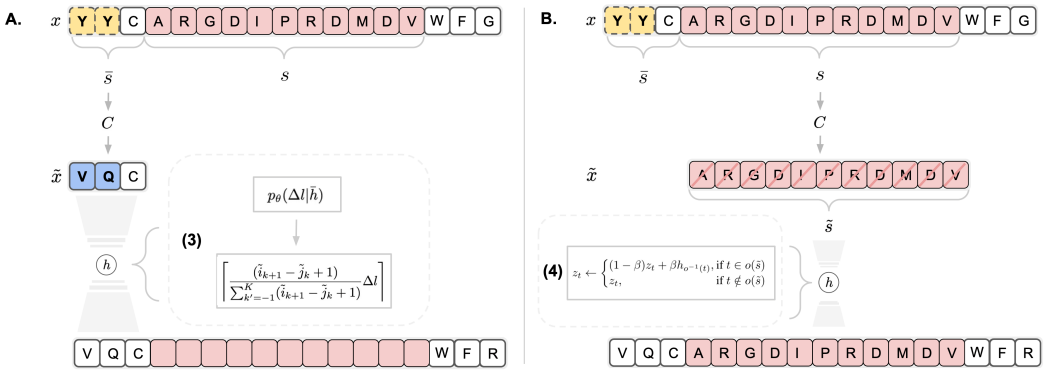

Figure 1: Multi-segment preserving sampling for $\beta = 1$. (A) Non-preserved segments $\bar{s}$ are corrupted using corruption process C, for which a given token (yellow) may be randomly perturbed (blue). This is encoded as hidden vector set h. Length change predictor $p_\theta(\Delta l|\bar{h})$ outputs Δl, which is distributed across $\bar{s}$ (Eq. 3). (B) Segment-preserving sampling follows similar operations on preserved segment s (red) with notable differences. Corruption C yields an unaltered sequence $\tilde{x}$ and we carry over hidden vector h_t of a token within preserved segment $\tilde{s}$ with strength β (Eq. 4).

At training time, we minimize the negative log-probability of the original sequence x given the corrupted version $\tilde{x}$ and a known Δl^* to train the encoder and decoder, while minimizing the negative log-probability of Δl^* to train the length change predictor. We parameterize the latter as a classifier. Once training is done, we can draw a series of samples from the deep manifold sampler by repeating the process of corruption, length conversion, and reconstruction.

While the original deep manifold sampler has an additional function predictor that can be used to guide the sampling procedure, we omit that here, as this is optional and can be replaced with another computational oracle without altering the sampling procedure that is the focus of this paper.

3 Multi-Segment Preserving Sampling

The deep manifold sampler was originally proposed in the context of protein design in Gligorijević et al. (2021). Within this setting, we often consider biological, chemical, and physical knowledge in order to impose constraints that narrow down a large, combinatorial search space (Street and Mayo, 1999; Woolfson, 2021). The deep manifold sampler on the other hand stays true to the key principle of deep learning, that is, end-to-end learning, which makes it challenging to explicitly incorporate this knowledge into both learning and sampling. In this paper, we take one step towards enabling this into the sampling procedure of the deep manifold sampler. We assume the availability of knowledge in which segments of an original sequence, from which sampling starts, must be preserved in order to maintain a set of desirable properties. For example, in the case of antibody engineering, it may be desirable to only alter CDR loops while leaving all framework residues intact (Kuroda et al., 2012).

Let $x = (x_1, x_2, \ldots, x_L)$ be the initial sequence from which we run the deep manifold sampler to draw a series of samples over the sequence manifold. Instead of unconstrained sampling, we consider a scenario in which we are provided with a set of non-overlapping segments of the sequence that must be preserved in-order by their starting and ending indices (inclusive):

$$s = ((i_1, j_1), \ldots, (i_K, j_K))$$

subject to $i_k \leq j_k$ for all k and $j_k < i_{k'}$ for all $k' > k$. We refer to this set as a *preserved-segment set*. Likewise, we can imagine the complement segment set $\bar{s}$ that contains all the segments that are between the to-be-preserved segments in s:

$$\bar{s} = ((0, i_1 - 1), (j_1 + 1, i_2 - 1), \ldots, (j_K + 1, |x| + 1)).$$

In order to preserve these segments while altering the remaining parts of the sequence, including their respective lengths, we make a series of modifications to the sampling procedure of the deep manifold sampler. First, we alter the corruption process C such that it does not corrupt the preserved segments. For instance, if the corruption process randomly adds or removes tokens, this is only done to the segments in the complement set $\bar{s}$ but not to those in s. Doing so results in $\tilde{x}$ and also changes the segment set s in order to appropriately reflect the changes in the indices due to insertions and deletions. We use $\tilde{s}$ to refer to the preserved segment set of the corrupted sequence $\tilde{x}$.

However, the encoder still results in the hidden vector set h that corresponds to $\tilde{x}$. While the length change prediction also stays the same, returning length change Δl, Δl needs to be distributed across the non-preserved segments in order to avoid altering the length of any preserved segment in $\tilde{s}$. We do so proportional to the original lengths of the non-preserved segments. In other words, we add to the length of each non-preserved segment $(\tilde{j}_k + 1, \tilde{i}_{k+1} - 1)$:

$$\left\lceil \frac{(\tilde{i}_{k+1} - \tilde{j}_k + 1)}{\sum_{k'=-1}^{K} (\tilde{i}_{k+1} - \tilde{j}_k + 1)} \Delta l \right\rceil, \tag{3}$$

where $\tilde{j}_0 = 0$ and $\tilde{i}_{K+1} = |\tilde{x}| + 1$.

After distributing the length difference among the non-preserved segments, we can now construct the index map o that tells us which segment in the new sequence corresponds to each of the preserved segment in $\tilde{x}$. In other words, $y_{o(\tilde{i}_k):o(\tilde{j}_k)} = \tilde{x}_{\tilde{i}_k:\tilde{j}_k}$. Let us use $o(\tilde{s})$ to denote the preserved-segment set derived from s and the length distribution above.

The actual length conversion happens just like before, as in Eq. (1). We however add an extra step after the length conversion in order to give the decoder a hint about preserved segments and their contents. This is done by carrying over the original hidden vector h_t of a token within a preserved segment:

$$z_t \leftarrow \begin{cases} (1-\beta) z_t + \beta h_{o^{-1}(t)}, & \text{if } t \in o(\tilde{s}) \\ z_t, & \text{if } t \notin o(\tilde{s}) \end{cases} \tag{4}$$

o^{-1} is the inverse index map, and $\beta \in [0, 1]$ is the strength of carry-over.

The decoder turns this length-converted and segment-preserving hidden sequence z into a sequence of logit vectors $\tilde{y}$, just like the original sampling procedure. We then modify the logit vector corresponding to a token with a preserved segment to force the sampled outcome to preserve the token identity:

$$\tilde{y}_t^v \leftarrow \begin{cases} \infty, & \text{if } t \in o(\tilde{s}) \text{ and } v = \tilde{x}_{o^{-1}(t)} \\ -\infty, & \text{if } t \in o(\tilde{s}) \text{ and } v \neq \tilde{x}_{o^{-1}(t)} \\ \tilde{y}_t^v, & \text{if } t \notin o(\tilde{s}) \end{cases} \tag{5}$$

In the case of non-autoregressive modeling, this would result in a Categorical distribution for a preserved token to assign the entire probability mass ($= 1$) to the original token identity. If a CRF is used at the end, this would prevent any sequence that violates preservation from being decoded out with non-zero probability.

We can repeat this sampling step with the newly sampled sequence and the corresponding preserved-segment set. This allows us to iteratively draw a series of samples while preserving the segments from the original sequence, designated by the preserved-segment set s. Because this iterative sampling procedure preserves multiple segments and their contents, we refer to this procedure as *multi-segment preserving sampling* (Figure 1).

4 Related Work

There are two alternative sequence-modeling paradigms that are closely related to the deep manifold sampler. We briefly describe each of them here in the context of whether and how multi-segment preservation can be implemented.

4.1 MASKED LANGUAGE MODELS

A masked language model is a special case of a DAE, similar to the deep manifold sampler (Devlin et al., 2018; Liu et al., 2019). A corruption process C in a masked language model is designed to apply one of three types of corruption: (1) replace a token with a special $\langle$mask$\rangle$ token, (2) replace a token with another, randomly-selected token, and (3) no alteration, to a random subset of the tokens within each sequence. All of these types do *not* alter the length of the sequence. The masked language model then reconstructs only the corrupted subset, rather than the full sequence as in the deep manifold sampler.

Although masked language modeling was not originally motivated as a generative model from which we can draw sequences, but instead as a way to pretrain a large-scale neural network, Wang and Cho (2019) and Goyal et al. (2021) demonstrated that it is possible to draw well-formed sequences from masked language models. Despite this, masked language modeling is not as popular nor applicable as a sequence generator as other approaches — such as autoregressive language models or the deep manifold sampler — because it does not have an inherent mechanism to automatically model the length distribution.

With the caveat that the length of a sequence cannot be altered at sampling time, a masked language model can be used for multi-segment preserving sampling. This can still be useful as a "plug-and-play" proposal distribution for a number of downstream tasks including rational design and directed evolution of proteins (Woolfson, 2021; Arnold, 1998; Yang et al., 2019; Meier et al., 2021). However, this approach is limited compared to the proposed strategy of multi-segment preserving sampling with the deep manifold sampler, as our proposal is able to both dynamically adapt the length of a sequence and preserve segments.

4.2 DENOISING SEQUENCE-TO-SEQUENCE MODELS

A denoising sequence-to-sequence (Seq2Seq) model, where the decoder is autoregressive, has been studied previously in the context of natural language processing (Hill et al., 2016; Lewis et al., 2019). Unlike masked language models, and similar to the deep manifold sampler, the denoising Seq2Seq model can adaptively change the length of a sequence. However, unlike the deep manifold sampler, there is less control over the direct manipulation of the intermediate hidden vectors, as some of the dependencies between the tokens in the output sequence are captured directly by the decoder without relying on the intermediate representation between the encoder and decoder. It is however an interesting future direction to compare the denoising Seq2Seq model against the deep manifold sampler.

With respect to multi-segment preservation, denoising Seq2Seq models do not readily admit such a sampling strategy. This is due to the inherent intractability in decoding from an autoregressive model with unbounded context. This intractability is often addressed by approximate decoding, such as beam search, which is known to have suboptimal behaviors despite its successful and wide use (Welleck et al., 2020; 2019). Several studies have proposed to extend beam search to incorporate such constraints (Hokamp and Liu, 2017; Post and Vilar, 2018). Unfortunately most of these approaches incur great computational cost, as their computational complexity grows often linearly with respect to the number of preserved segments — or the beam size must grow accordingly — because the underlying algorithm decodes in a greedy left-to-right fashion with a limited hypothesis set (beam). While it is possible to modify a denoising Seq2Seq model to admit multi-segment preserving sampling by letting the decoder only reconstruct non-preserved segments,this is out of scope for this paper.

5 EXPERIMENTS

The proposed algorithm for multi-segment preserving sampling is designed to completely preserve designated segments. Here, we demonstrate a potential application in antibody design enabled by our algorithm coupled with the deep manifold sampler. Antibodies with a particular V-gene have fixed lengths in the framework as well as CDR1 and CDR2 regions. As a result, antibodies display most of their diversity in length and amino acid composition in CDR3 (Glanville et al., 2009). To demonstrate the effectiveness of our approach and restricted variation of the preserved segments, we

β	**Aligned CDR3 sequence**	**Edit distance**
N/A (original)	ARDPEWDPF-QANY-YYYGMDV	0
0.0	ARDPEWDPF-QAN--YYYGMDV	3
0.1	ARDPEWDPFFQANYNYYYGMVD	3
0.5	KRDPEWDRF-QAPY-YTVGMDV	5
0.9	ARGPECDPH-QAV-DIYYGMDV	6

Table 1: Example outputs of multi-segment preserving sampling when restricting variation to the CDR3 region under different settings of β. Display is restricted to the sampled region, the rest is preserved by construction.

select all unique human antibody sequences with the *IGHV1-18* gene from the Observed Antibody Space (OAS) database (Olsen et al., 2022) for multi-segment preserving sampling. Using a deep manifold sampler, we sample exclusively from the CDR3, while preserving other regions, and show the length and log-probability (GPT-2) distributions of the generated sequences qualitatively coincide with that of the test data. Table 1 illustrates examples of sampled CDR3 regions under different settings of carry-over strength β.

5.1 Training details

We obtained 5,971,552 unique human antibody heavy chain sequences with the *IGHV1-18* gene from the OAS database, with 2,000 and 10,000 sequences set aside for validation and test sets respectively and the remaining used for training.[1] We trained a deep manifold sampler on the training set with a constant learning rate of 10^{-4} for 60K mini-batch steps with the batch size of 128. The model consisted of a two-layer transformer encoder and decoder, each with 8 heads and the total embedding dimension of 256 and feed-forward layer dimension of 1024. The last layer consists of a CRF for final sequence generation. The rest of the training procedure was the same as described in the original paper.

In addition, we also trained an autoregressive GPT-2 model using HuggingFace Transformers library v4.16.2 (Wolf et al., 2020) on the same training set in order to demonstrate that the sampler-generated sequences capture the amino-acid token distribution observed in the training set. The model consisted of 6 attention layers with 8 heads and a total embedding dimension of 512 and was trained with a constant learning rate of 4×10^{-4} for 25K mini-batch steps with batch size of 1024. The other parameters were set to the default values provided by the package.

5.2 Sampling details and results

For each sequence in the test set, we applied multi-segment preserving sampling for one iteration, preserving all non-CDR3 regions with four different β values of 0, 0.1, 0.5, and 0.9.[2]

Figure 2 and Figure 3 show the length and log-probability (GPT-2) distributions of the generated sequences with changes in CDR3 and the test data across all selected β values. The CDR3 length distribution of the generated samples matches the natural sequence length distribution for each value of β. The GPT-2 log-probability distribution of the samples has lower overall mean compared to that of the test distribution but is still within the same range. Both distributions vary only slightly with different values of β. These two results show the effectiveness of the sampling strategy for generating diverse antibody sequences, restricted to user-defined regions.

In Figure 4, we illustrate the distribution of the number of edits in the generated sequences relative to the input seed sequences, including substitutions, insertions, and deletions. The distributional mean increases slightly with higher values of β. For future work, we plan on a more systematic understanding of the effects of carry-over strength β on sample quality and diversity.

[1] Only sequences with "Redundancy > 1" were retained.

[2] The region annotation was obtained from the OAS data unit files.

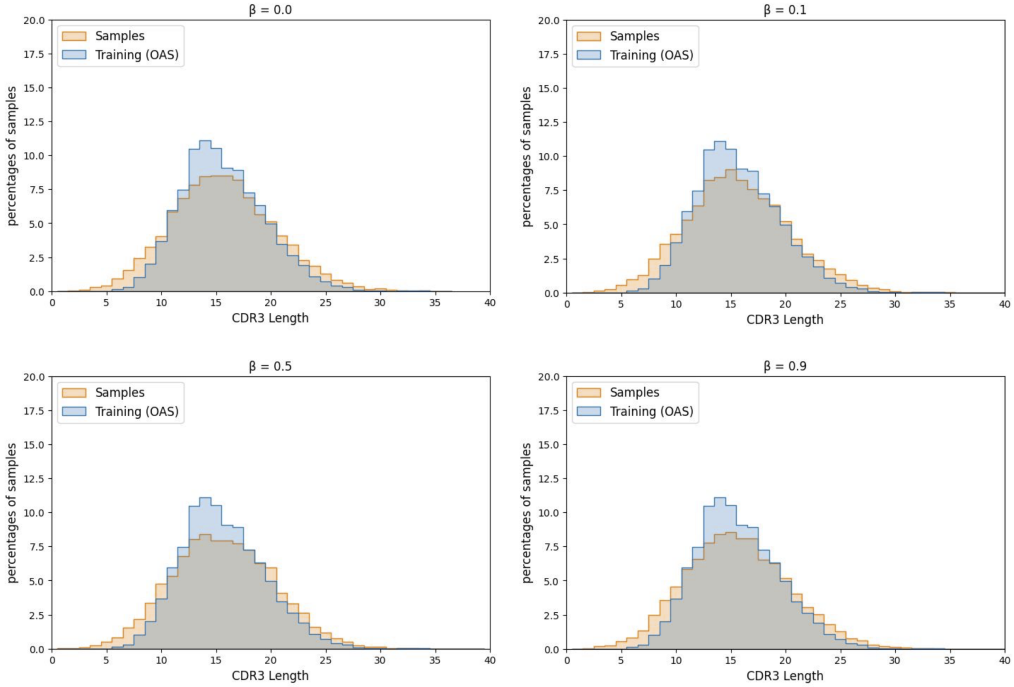

Figure 2: The normalized distribution of the CDR3 lengths of the deep manifold sampler-generated sequences ("Samples") and the test set sequences ("Training (OAS)") with four different β parameters. From Top Left, Clockwise: Samples were generated with $\beta = 0, 0.1, 0.9$, and 0.5.

6 Conclusion

We have proposed a sampling procedure for the deep manifold sampler that explicitly preserves designated segments of the input sequence, allowing variation to occur only in non-preserved regions. We find that this approach, multi-segment preserving sampling, is applicable to a number of design problems in the life sciences where we often want to use prior knowledge made available in these domains. With biological sequence design for example, we want to sample new, diverse designs that avoid perturbing well-conserved regions of the input. In this way, we demonstrate the utility of multi-segment preserving sampling by restricting sampling to the CDR3 regions of a collection of antibody heavy chains with the *IGHV1-18* gene and validating the resulting CDR3 designs against a separate GPT-2 model. As shown in Figure 3, the sampled CDR3 regions admit high log-probability scores by the GPT-2 model, providing evidence that the samples are plausible. Additionally, the CDR3 length distribution of the samples coincides with the observed length distribution in the training set, suggesting the model adequately captures the variability in non-preserved segment lengths, despite the lack of explicit provision during training. In future work, we will extend our exploration on the effect of the carry-over strength β in terms of sample quality as well as its usage in conjunction with the function predictor for guided sampling proposed in Gligorijević et al. (2021).

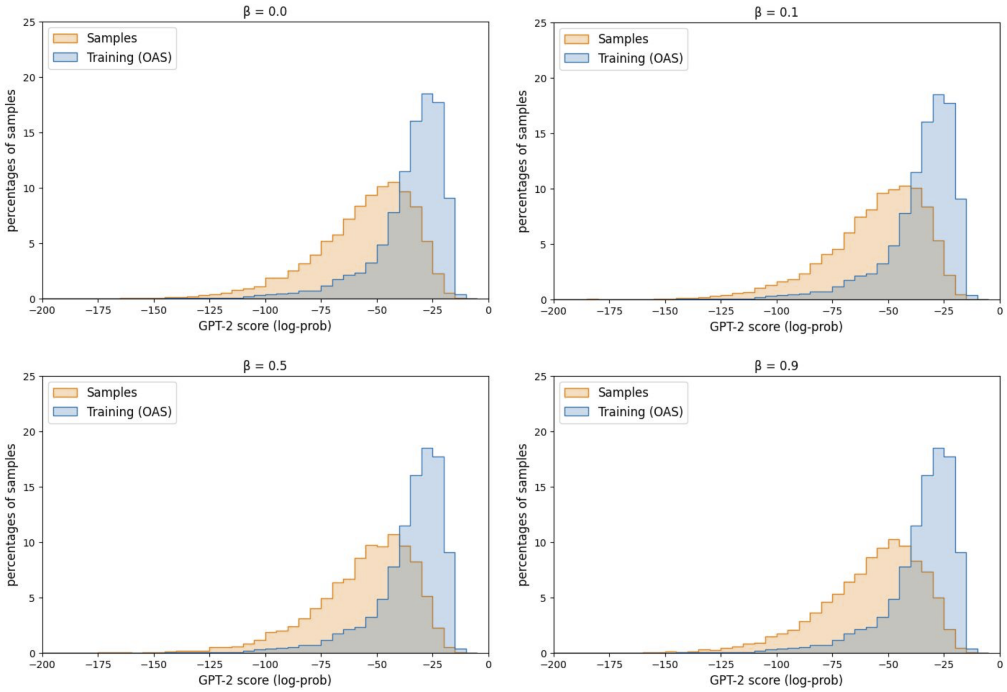

Figure 3: The normalized distribution of the GPT-2 scores of the deep manifold sampler-generated sequences ("Samples") and the test set sequences ("Training (OAS)") with four different β parameters. From Top Left, Clockwise: Samples were generated with $\beta = 0, 0.1, 0.9$, and 0.5.

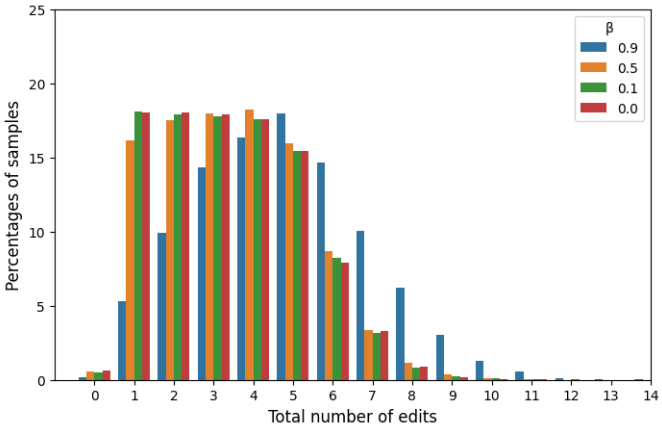

Figure 4: The distribution of edit distances between generated samples and their seed sequences with varying settings of β (0, 0.1, 0.5, and 0.9).

REFERENCES

Frances H Arnold. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Mark L Chiu, Dennis R Goulet, Alexey Teplyakov, and Gary L Gilliland. Antibody structure and function: the basis for engineering therapeutics. *Antibodies*, 8(4):55, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

James Dunbar and Charlotte M. Deane. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 09 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv552. URL https://doi.org/10.1093/bioinformatics/btv552.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing, 2021.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, 2017.

Jacob Glanville, Wenwu Zhai, Jan Berka, Dilduz Telman, Gabriella Huerta, Gautam R Mehta, Irene Ni, Li Mei, Purnima D Sundar, Giles MR Day, et al. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proceedings of the National Academy of Sciences*, 106(48):20216–20221, 2009.

Vladimir Gligorijević, Daniel Berenberg, Stephen Ra, Andrew Watkins, Simon Kelow, Kyunghyun Cho, and Richard Bonneau. Function-guided protein design by deep manifold sampling. *bioRxiv*, 2021.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. Exposing the implicit energy networks behind masked language models via metropolis–hastings. *arXiv preprint arXiv:2106.02736*, 2021.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*, 2016.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*, 2017.

Simon P. Kelow, Jared Adolf-Bryfogle, and Roland L. Dunbrack. Hiding in plain sight: structure and sequence analysis reveals the importance of the antibody de loop for antibody-antigen binding. *mAbs*, 12(1):1840005, 2020. doi: 10.1080/19420862.2020.1840005. URL https://doi.org/10.1080/19420862.2020.1840005. PMID: 33180672.

Daisuke Kuroda, Hiroki Shirai, Matthew P Jacobson, and Haruki Nakamura. Computer-aided antibody design. *Protein engineering, design & selection*, 25(10):507–522, 2012.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*, 2018.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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, 2020.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34, 2021.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*, 2018.

Martin Reczko, Andrew CR Martin, Henrik Bohr, and Sándor Suhai. Prediction of hypervariable cdr-h3 loop structures in antibodies. *Protein Engineering, Design and Selection*, 8(4):389–395, 1995.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/content/118/15/e2016239118`.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8846–8853, 2020.

Arthur G Street and Stephen L Mayo. Computational protein design. *Structure*, 7(5):R105–R109, 1999.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*, 2019.

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.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. Consistency of a recurrent language model with respect to incomplete decoding. *arXiv preprint arXiv:2002.02492*, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Derek N Woolfson. A brief history of de novo protein design: minimal, rational, and computational. *Journal of Molecular Biology*, 433(20):167160, 2021.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

Ren Yi, Kyunghyun Cho, and Richard Bonneau. Nettime: Improving multitask transcription factor binding site prediction with base-pair resolution. *bioRxiv*, 2021.