Compressing the Latent Space of Single-Sequence Protein Predictors for Multimodal Generation

Anonymous Authors[1]

Abstract

ESMFold learns a joint latent space of sequence and structure while *requiring only sequence as input*. However, the latent space of ESMFold is disorganized and we find pathologies, similar to those observed in large language models, that render these models unusable for multimodal representation learning. Meanwhile, latent diffusion in both continuous and discrete spaces have improved efficiency and performance in image and multimodal generation, but are built on an abundance of knowledge on autoencoders for images. To create a protein encoder which captures structural and functional information for generative modeling in the latent space, we create CHEAP (Compressed Hourglass Embedding Adaptations of Proteins) representations, and find that the channel dimension of ESMFold latent spaces can be compressed by up to $256\times$ while retaining rich structural, sequence, and functional information, as demonstrated on protein understanding benchmarks and reconstruction performance.

1. Introduction

Generative modeling has emerged as a popular tool for protein design due to its scaling properties on complex data distributions (Watson et al., 2023; Ingraham et al., 2022). To synthesize the molecule in the lab, however, a sequence that can fold into the structure must be specified. Despite the dual importance of structure and sequence modalities, existing methods are typically single modality, and generate either structure (Watson et al., 2023; Ingraham et al., 2022) or sequence (Gruver et al., 2023; Alamdari et al., 2023). By sampling sequence and structure simultaneously, one gains structure-conditioned control over protein design, which is highly useful given the large array of structure-mediated use cases in drug discovery, such as efficient hit binding, targeting specific biological pathways, learning protein-protein interactions (PPI), and perform more efficient docking.

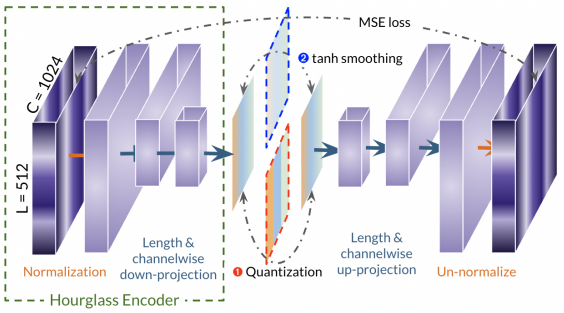

Figure 1. Overview of the compression scheme. The protein language model output contains massive activations, and is first normalized using the statistics of each channel. Then, the Hourglass Encoder architecture is used, where linear projections are used to shorten along the length dimension and downproject along the channel dimension. In the bottleneck layer, we examine methods for obtaining both discrete and continuous compressed embeddings, as described in Section 3.2.

Sequence-to-structure prediction (Jumper et al., 2021; Lin et al., 2023) have been posited as "protein structure foundation models" (Wang et al., 2024). ESMFold (Lin et al., 2023) demonstrates that sequence-to-structure prediction can be built on top of protein language model (pLM) embeddings. Intriguingly, at inference time, though pLM attentions capture pairwise contact information (Rao et al., 2021), the pairwise input to the structural module is initialized to zero (Section C and Figure 2). All the information required for the structure, therefore, is contained within this sequence embedding. Thus, by learning a generative model to approximate the distribution of natural proteins under this representation, one can perform simultaneous multimodal generation of structure and sequence. Importantly, this allows structural diffusion from *only sequence as inputs*, which is desirable because sequence datasets can be 10^2 to 10^4 times larger than structural datasets.

Naively intercepting this latent space, however, presents numerous challenges. The latent space of large language models (LLMs) often have high activations in certain chan-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Accessible and Efficient Foundation Models for Biological Discovery. Do not distribute.

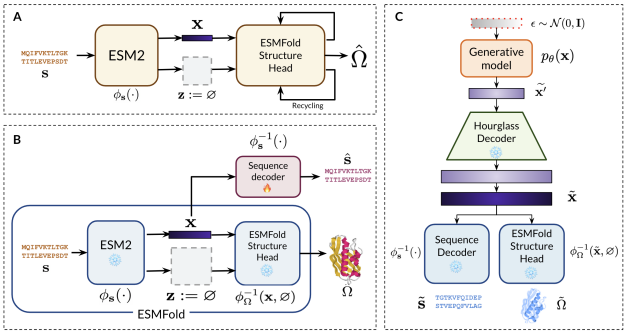

Figure 2. Learning the joint distribution of protein sequence and structure as the latent space of ESMFold for multimodal generation. **(A)** Overview of the ESMFold (Lin et al., 2023) model at inference time. **(B)** Disassembling ESMFold for latent multimodal generation. By training a decoder from $\mathbf{x}$ back to sequence, and using the pretrained ESMFold Structure Head, we obtain deterministic mappings between $\mathbf{x}$ and both the sequence and structure spaces. **(C)** At inference time, given a learned generative model $p_\theta(\mathbf{x})$, we can sample compressed latent $\widetilde{\mathbf{x}}'$ embeddings, decompress them (see Section 3.1, and map them back to sequence and structure, thus simultaneously generating both structure and sequence.

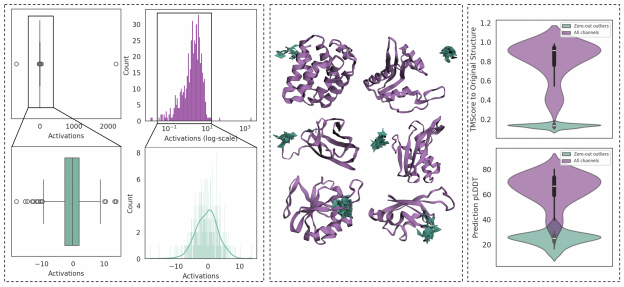

Figure 3. **(Left)** Histogram of per-channel means, and after removing three outlier channels with mean absolute values >20. **(Middle)** Original prediction (purple) entirely deteriorates after setting these three outlier channels to zero (teal). **(Right)** Model performance deteriorates after dropping outlier channels, as quantified by the TMScore (structure accuracy) and pLDDT (model confidence).

nels that persist regardless of the input sequence (Sun et al., 2024) (Figure **??**), rending them unwieldy for learning with a generative model. Furthermore, the large dimensionality of language model embeddings render them difficult to learn. The intrinsic dimensionality of protein language model is often much smaller than the actual channel dimension (Valeriani et al., 2024), suggesting that they can maybe be compressedto smaller dimensions while retaining the sturctural and functional information desirable for protein design.

Contributions Towards our goal of taming the latent space of sequence-to-structure models for flexible, controllable, and compute-efficient latent generation for proteins, we compress the ESMFold latent space and introduce **CHEAP** (Compressed Hourglass Embedding Adaptations of Proteins) representations. CHEAP embeddings are designed for latent generation but also perform competitively on function, localization, and structure-related benchmarks. We also demonstrate how compression affects reconstruction performance and function prediction across both discrete and continuous compression schemes.

2. Related Works

Latent Space Generation in Visual Media Latent-space based generative models is often used to manage the high-dimensional nature of visual data; design of these successful methods is built on ample research around architectural and algorithmic choices for visual representations. Contemporary scalable generative models for vision and multimodal media often fall into two categories: those working with discrete representations in either an masked-token or autoregressive next-token prediction manner (He et al., 2022; Bao et al., 2021; Chang et al., 2022; Yu et al., 2022; Razavi et al., 2019; Villegas et al., 2022; Esser et al., 2021), or diffusion-based models with continuous data (Ho et al., 2020; Saharia et al., 2022; Rombach et al., 2022; Peebles & Xie, 2023; Ho et al., 2022).

Protein diffusion and multimodal generation Though protein structure diffusion has seen empirical and lab-verified success (Watson et al., 2023; Bennett et al., 2024), such models learn a probability distribution over plausible protein structures, rather than the joint distribution of both sequence and structure. Such models rely on an exogenous structure-to-sequence prediction step to obtain the sequence. Empirical results show that such methods often exhibit "low designability", where generated structures may not have a sequence that can fold into that structure. Some works attempt to generate both structure and sequence simultaneously, usually alternating between sequence-to-structure and structure-to-sequence steps (Lisanza et al., 2023; Chu et al., 2023).

Massive Activations in LLMs Large transformers often suffer from the *massive activations* (Sun et al., 2024) or *outlier features* (Dettmers et al., 2022) phenomenon, where output values in intermediate layers exhibit unusually high values on the magnitude of up to 20x larger. Sun et al. (2024) provides detail study and finds that for both Llama and ViT, finding channels which have outlier values, and dominate attention patterns. In contrast to the well-tamed latent space of two-stage latent diffusion works in images (Rombach et al., 2022), the latent space of LLMs should be expected to be much more unwieldy, and we indeedly find this to be true empirically (Figures 2).

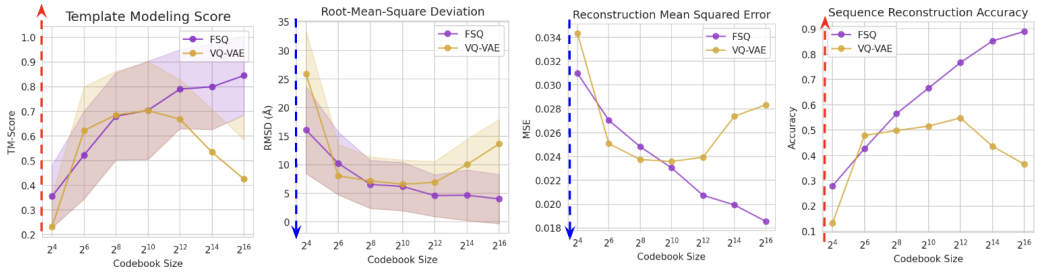

Figure 4. Comparing FSQ and VQ-VAE performance across a range of codebook sizes. Consistent with reported findings in Mentzer et al. (2023), performance is initially higher for VQ-VAE, with FSQ outperforming VQ-VAE at codebook sizes greater than 2^10. Blue arrows denote metrics where lower is better, and red arrows denote metrics where higher is better.

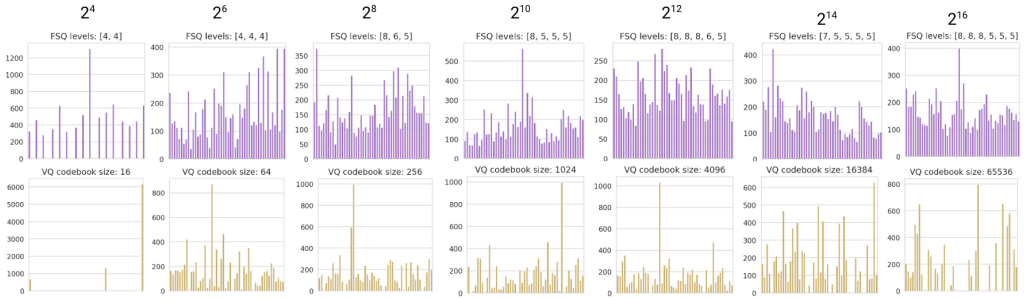

Figure 5. Examining codebook utilization for FSQ and VQVAE. VQVAE suffers from codebook collapse, while FSQ is constructed by design to use all codes. Codebook utilization is generally more favorable wih FSQ (top).

3. Methods

3.1. Organizing the Latent Space for Generation

Per-Channel Normalization To address the issue of massive activations as shown in Figure 2, we use a per-channel normalization scheme. The embeddings are processed as:

$$\mathbf{x}' = \frac{\mathbf{x} - \mathbf{x}_{\min}}{\mathbf{x}_{\max} - \mathbf{x}_{\min}} \times \Big((c_{\max} - c_{\min}) + c_{\min}\Big)$$

where $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are vectors with shape $(1024,)$ broadcasted along the length dimension to match $\mathbf{x}$, and denote statistics calculated for each channel, independently. This prevents outlier channel values from dominating the normalization. For consistency with image diffusion works, we choose $c_{\min} = -1$ and $c_{\max} = 1$.

Latent Space Compression with the Hourglass Compression Transformer Unlike images, proteins have different lengths, which precludes usage of convolution-based autoencoders. However, we reason that the downsampling operation in convolution neural network may also be key for compressing information from adjacent amino acids into local motifs. Furthermore, on transformer-based generative models, reducing the length dimension also helps with managing the quadratic memory requirements of transformer attention layers (Vaswani et al., 2017). Therefore, we choose an encoder architecture inspired by the Hourglass Transformer (Nawrot et al., 2021), which includes a shortening operation, $g(x)$, that transforms a tensor x with shape (L, D) to $(\frac{L}{S}, D)$. The Hourglass Compression Transformer architecture is described in Algorithm A.

3.2. Compression Representations

For discrete representation, we further examine two schemes: (1) Vector-quantized variational auto-encoders (VQ-VAE) (Van Den Oord et al., 2017) and (2) Finite Scalar Quantization (FSQ) (Mentzer et al., 2023). The VQ-VAE (Van Den Oord et al., 2017) learns a discrete representation of the input, typically of images. In the forward pass, the encoder h_e produces a continuous feature representation of input $\mathbf{x}$. Then, each feature vector is mapped to a discrete code in the codebook space, $\mathcal{C}$, where each discrete code is associated with a continous vector $\mathbf{e}_i$. The complete VQ-VAE loss is:

$$L_{\text{VQ}} = \log p(\mathbf{x}|h_q(\mathbf{z})) + ||\text{sg}[h_e(\mathbf{x})] - \mathbf{z}||_2^2 + \beta||h_e(\mathbf{x}) - \text{sg}[\mathbf{z}]||_2^2$$

VQ-VAE can be prone to “codebook collapse", whereby a few codes are over-utilized, especially for larger codebook sizes (Takida et al., 2022; Łańcucki et al., 2020; Dhariwal et al., 2020; Huh et al., 2023). We therefore also investigate using the FSQ (Mentzer et al., 2023) approach. Rather than using a nearest-neighbor search to choose a code, FSQ directly quantizes the continuous encoder representations $\mathbf{z} \in \mathbb{R}^d$ into L bins:

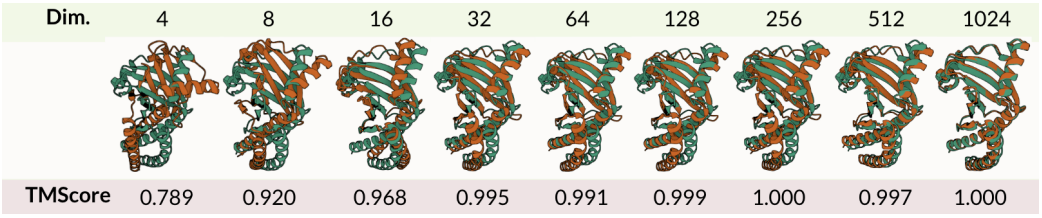

Figure 6. Qualitative examination of structure reconstruction $\phi_\Omega^- 1(\mathbf{x}_{\text{reconstructed}})$, for different bottleneck $\mathbf{c}$ dimensions (original embedding contains 1024 channels), and a **shortening factor of 2**. Despite aggressive downsampling and channel down projection, substantial structure information can be recovered via neural compression.

$$\begin{aligned} \mathbf{z} = h_e(\mathbf{x}), \mathbf{z} \in \mathbb{R}^d & \quad \text{Encoder output} \\ \hat{\mathbf{z}} = \tanh(\mathbf{z}) & \quad \text{Bound to } [-1, 1] \\ \hat{\mathbf{z}} = \text{round}(\lfloor (L/2) \rfloor \cdot \hat{\mathbf{z}}) & \quad \text{Discretize to } L \text{ bins} \end{aligned}$$

The predetermined bins L is selected to be small relative to VQ-VAE codebook sizes. The *implicit codebook size* $|\mathcal{C}|$, however, comes from the combinatorial possibilities arising from using one of L integers at each of the D channels. For $\mathbf{z}$ with d channels, there are d associated integer representations, and thus $|\mathcal{C}| = L^d$. A large implicit codebook can thus be achieved, while forcing all codes to be used.

4. Results

Similar to using perceptual loss evaluation in addition to reconstruction performance, we also examine reconstruction performance in sequence and structure space (Figure 2). Template-modeling score (**TM-Score**) is a backbone only metric of structure reconstruction, while root-mean-square deviation (**RMSD**) is a more fine-grained measure between atom positions. Sequence reconstruction accuracy examines token matches after decoding back to sequence space.

4.1. Discrete Compression

Though reference experiments exist in images with regard to how big codebook sizes should be, it is unclear how many bits of information can be expected from a joint representation of both sequence and all-atom structure. We therefore do a thorough investigation across different codebook sizes for both VQVAE and FSQ. Consistent with findings in Mentzer et al. (2023), we find that **FSQ outperforms VQVAE for codebook sizes larger than** 2^{10} across both reconstruction MSE and performance measured in structure and sequence spaces.

4.2. Continuous Compression

Table 4.2 examines performance on benchmarks from Xu et al. (2022). The downprojected version of the latent that is intercepted upstream of the original ESM output performs competitively or better than ESM1b, despite aggressive compression. More benchmark results can be found in the Appendix. Figure 4.2 demonstrates that good backbone alignment (i.e. TM-Score), RMSD below idealized inter-residue bond lengths, and near-perfect sequence reconstruction performance can be retained even after aggressive compression.

	# Dimensions	Cont	Fold	SSP	Yst
DDE	400	–	0.10	–	0.56
Moran	240	–	0.07	–	0.53
LSTM	640	0.26	0.08	0.69	0.54
Transformer	512	0.18	0.09	0.60	0.54
CNN	21	0.10	0.11	0.66	0.55
ResNet	512	0.20	0.09	0.70	0.49
ProtBert	1024	0.40	0.11	0.82	0.54
ESM-1b	1280	**0.46**	0.30	0.83	**0.66**
CHEAP (ours)	8	0.28	0.15	0.82	0.45
	64	0.42	0.45	0.85	0.48
	128	0.38	0.47	0.85	0.51
	256	0.23	0.50	0.85	0.51
	512	0.37	**0.53**	**0.86**	0.46

Table 1. Comparing representation learning results on benchmarks described in Xu et al. (2022). CHEAP performs competitively or better despite aggressive compression.

5. Conclusion

CHEAP embeddings investigate the compression of the ESMFold latent space for protein multimodal generation, to both enable speed and flexibility. Using an Hourglass autoencoder architecture, our results demonstrate that functional and structural information learned by ESMFold can be compactly captured.

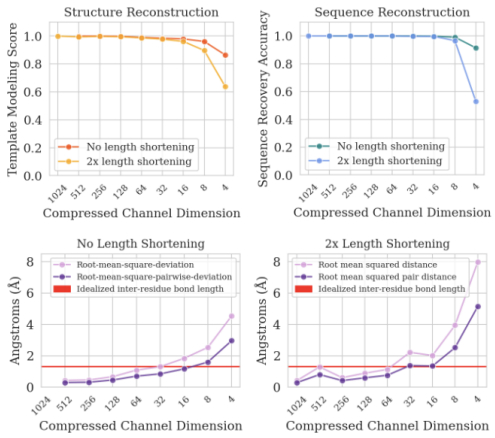

Figure 7. **(Top)** Comparing TM-Score and sequence recovery of compressed structure and the original prediction for compressed representations. **(Bottom)** Comparing the RMSD with RMSPD (super-imposition free) to distinguish reconstruction errors in orientation only vs. those which also alter pairwise distances.

A. Hourglass Compression Transformer

Though different operations may be used for $g(x)$, we find a simple linear downsampling to work well (Algorithm **??**), which can be seen as a convolution with a filter size and stride both equal to S. Since the original model is designed for sequence-to-sequence tasks rather than compression, we remove the skip connections that would make the solving the reconstruction task trivial. Additionally, we add a projection layer along the channel dimension after each shortening operation. At training time, $\mathbf{x}_{\text{reconstructed}}$ is used for calculating the mean-squared-error reconstruction loss, and at inference time, the output of the encoder is used as the compressed representation, with additional processing in the bottleneck, depending on if the compression is discrete or continuous (Section 3.2).

Algorithm 1 Hourglass Compression Transformer

embedding $\mathbf{a} \leftarrow \mathbf{a} \in \mathbb{R}^{L \times D}$,
mask $\mathbf{m} \leftarrow \mathbf{m}^{\text{Length}[L]} := \{1, 0\}_l^L$,
shortening factor $S \leftarrow S \in \mathbb{Z}$,
downprojection factor $K \leftarrow K \in \mathbb{Z}$,
downprojection $W_d \leftarrow W_d \in \mathbb{R}^{D \times \frac{D}{K}}$,
upprojection $W_u \leftarrow W_u \in \mathbb{R}^{\frac{D}{K} \times D}$
$\mathbf{a}, \mathbf{m} \leftarrow$ Pad length to multiple of S
$\mathbf{a} \leftarrow$ Transformer($\mathbf{a}$, $\mathbf{m}$)
$\mathbf{a}' \leftarrow$ LinearDownsampling($\mathbf{a}, S$)
$\mathbf{m}' \leftarrow \sum_S \mathbf{m}^{\text{Length}[L] \rightarrow [\frac{L}{S}]} > 0$ {*Reduce*}
$\mathbf{a}' \leftarrow$ AttentionResampling($\mathbf{a}', \mathbf{a}, \mathbf{m}'$)
$\mathbf{c} \leftarrow W_d \mathbf{a}'$
if quantize **then**
  $\mathbf{c} \leftarrow$ Bottleneck($\mathbf{c}, \mathbf{m}'$)
else
  $\mathbf{c} \leftarrow$ Tanh($\mathbf{c}$)
end if
$\mathbf{a}' \leftarrow W_u \mathbf{c}$
$\mathbf{a} \leftarrow$ LinearUpsampling($\mathbf{c}', \mathbf{m}'$)
$\mathbf{m} \leftarrow \mathbf{m}'^{\text{Length}[\frac{L}{S}] \rightarrow [L]} > 0$ {*Repeat*}
$\mathbf{a} \leftarrow$ AttentionResampling($\mathbf{a}, \mathbf{a}', \mathbf{m}$)
$\mathbf{a}_{\text{reconstructed}} \leftarrow$ Transformer($\mathbf{a}$, $\mathbf{m}$)
return: $\mathbf{a}_{\text{reconstruction}} \in \mathbb{R}^{L \times D}, \mathbf{c} \in \mathbb{R}^{\frac{L}{S} \times \frac{D}{K}}$

B. Discrete Representation Learning

The assembled array of learned codes and their vector embeddings $\mathbf{z} = \{\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_{|\mathcal{C}|}\}$ and their corresponding feature features are fed into the decoder $h_q(\mathbf{z})$. Since the quantization operation is not differentiable, the straight-through estimator (STE) () is used by copying the gradients from the decoder input to the encoder output. The codebook is selected via a nearest-neighbor search in Euclidean space; auxiliary losses are introduced to pull the codeword vectors towards the unquantized encoder outputs. As in autoencoder training, a reconstruction loss between output and input is also used.

C. Defining a Joint Structure-Sequence Latent Space

A key observation for this work is that during inference use, it is empirically sufficient to initialize the pairwise representation input $\mathbf{z}$ as an array of zeros, and thus all information needed for structure is contained in $\mathbf{x} = \phi_{\mathbf{s}}(\mathbf{s})$ (Figure 2). The core idea of the PLAID framework is to train a generative model $p_\theta(\mathbf{x})$ to characterize the joint latent space of all feasible protein sequence and structures as $\phi(\mathbf{s}, \Omega)$, as defined by the intermediate layers of ESMFold (Lin et al., 2023). We intercept the sequence representation that is the direct input into the folding trunk (Figure 2).

Constructing forward- and backward-mappings To define mappings from sequence $\mathbf{s}$ and structure Ω to the joint multimodal representation space $\phi(\mathbf{s}, \Omega)$, we can decompose $\phi(\mathbf{s}, \Omega) = \phi_{\mathbf{s}}(\cdot) \circ \phi_\Omega(\cdot)$, and use components of ESMFold to

represent $\phi_\Omega(\cdot)$ and $\phi_{\mathbf{s}}(\cdot)$ mappings:

$$\mathbf{x} = \phi_{\mathbf{s}}(\mathbf{s}) \qquad \text{ESM2 Language Model} \tag{1}$$

$$\Omega = \phi_\Omega^{-1}(\mathbf{x}, \varnothing) \qquad \text{ESMFold Structure Module} \tag{2}$$

At inference time, after sampling $\tilde{\mathbf{x}} \sim p_\theta(\mathbf{x}) = p_\theta(s, \Omega)$, we can generate new protein sequences as $\tilde{\mathbf{s}} = \phi_{\mathbf{s}}^{-1}(\tilde{\mathbf{x}})$, which we see from Eq. 1 is the backward mapping of the ESM2 language model. This "back-mapping" sequence decoder can be trained separately, and since the space is already the output of a language model, we observe a per-token accuracy of 99.7% on a randomly partitioned heldout set.

D. Further Benchmark Results

	# Dimensions	Flu ↑	Sta ↑	β-lac ↑	Sol ↑	Sub ↑	Bin ↑
DDE	400	0.64	0.65	**0.62**	0.60	0.49	0.77
Moran	240	0.40	0.32	0.38	0.58	0.31	0.56
LSTM	640	0.49	0.53	0.14	0.70	0.63	0.88
Transformer	512	0.64	0.65	0.26	0.70	0.56	0.76
CNN	21	**0.68**	0.64	0.78	0.64	0.59	0.83
ResNet	512	0.64	0.13	0.15	0.67	0.52	0.79
ProtBert	1024	0.34	0.70	**0.62**	0.59	0.59	0.82
ESM-1b	1280	0.43	**0.75**	0.53	0.67	**0.80**	0.92
CHEAP (ours)	4	0.14	0.40	0.13	0.60	0.33	0.68
	8	0.22	0.44	0.17	0.64	0.45	0.74
	16	0.27	0.55	0.23	0.65	0.54	0.84
	32	0.28	0.56	0.28	0.67	0.57	0.87
	64	0.31	0.56	0.28	0.69	0.62	0.90
	128	0.41	0.58	0.38	0.70	0.68	0.90
	256	0.47	0.60	0.41	0.71	0.72	0.92
	512	0.51	0.63	0.36	**0.72**	0.74	0.93
No compression	1024	0.52	0.64	0.45	**0.72**	0.76	**0.94**

Table 2. Benchmarks on function and localization.

References

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pp. 2023–09, 2023.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Bennett, N. R., Watson, J. L., Ragotte, R. J., Borst, A. J., See, D. L., Weidle, C., Biswas, R., Shrock, E. L., Leung, P. J., Huang, B., et al. Atomically accurate de novo design of single-domain antibodies. *bioRxiv*, pp. 2024–03, 2024.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

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

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Gruver, N., Stanton, S., Frey, N. C., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. *arXiv*, 2305.20009, 2023.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Huh, M., Cheung, B., Agrawal, P., and Isola, P. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*, pp. 14096–14113. PMLR, 2023.

Ingraham, J., Baranov, M., Costello, Z., Frappier, V., Ismail, A., Tie, S., Wang, W., Xue, V., Obermeyer, F., Beam, A., et al. Illuminating protein space with a programmable generative model. *bioRxiv*, 2022.12.01.518682, 2022.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Łańcucki, A., Chorowski, J., Sanchez, G., Marxer, R., Chen, N., Dolfing, H. J., Khurana, S., Alumäe, T., and Laurent, A. Robust training of vector quantized bottleneck models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

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

Lisanza, S. L., Gershon, J. M., Tipps, S. W. K., Arnoldt, L., Hendel, S., Sims, J. N., Li, X., and Baker, D. Joint generation of protein sequence and structure with RoseTTAFold sequence space diffusion. *bioRxiv*, 2023.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Nawrot, P., Tworkowski, S., Tyrolski, M., Kaiser, Ł., Wu, Y., Szegedy, C., and Michalewski, H. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. MSA Transformer. *Proceedings of the 38th International Conference on Machine Learning*, 139:8844–8856, 2021.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

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

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Sun, M., Chen, X., Kolter, J. Z., and Liu, Z. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Takida, Y., Shibuya, T., Liao, W., Lai, C.-H., Ohmura, J., Uesaka, T., Murata, N., Takahashi, S., Kumakura, T., and Mitsufuji, Y. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. *arXiv preprint arXiv:2205.07547*, 2022.

Valeriani, L., Doimo, D., Cuturello, F., Laio, A., Ansuini, A., and Cazzaniga, A. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36, 2024.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv*, 1706.03762, 2017.

Villegas, R., Babaeizadeh, M., Kindermans, P.-J., Moraldo, H., Zhang, H., Saffar, M. T., Castro, S., Kunze, J., and Erhan, D. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.

Wang, J., Watson, J. L., and Lisanza, S. L. Protein design using structure-prediction networks: Alphafold and rosettafold as protein structure foundation models. *Cold Spring Harbor Perspectives in Biology*, pp. a041472, 2024.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620:1089–1100, 2023.

Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Chang, M., Liu, R., and Tang, J. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.