EXTRACTING AND STEERING ENSEMBLE AND STRUCTURAL PROPERTIES IN BIOEMU

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative protein models have recently demonstrated strong performance in protein structure prediction tasks. In particular, the AlphaFold models are the standard for static structure prediction. Other models like BioEmu add additional functionality by generating protein conformations trained to sample thermally distributed conformations. However, we still lack a mechanistic understanding of how these models sample the distribution of structures. To improve our underlying understanding of these models, we train sparse autoencoders on BioEmu to identify interpretable features corresponding to structural and thermodynamic properties. In addition, we assess at what point in the model these properties are most highly represented. From this, we determine that the hidden activations can be used to predict desired ensemble properties from a single sampling trajectory. Further, we show that the hidden activations can be directly tuned to steer generation towards a desired property, as well as towards specific conformational domains like cryptic pocket opening, domain motion conformers, and kinase states. These results improve mechanistic understanding of protein models and open the door for targeted property and conformer steering of protein structure sampling models like BioEmu.

1 INTRODUCTION

Protein conformations are highly varied, and many important conformations are rarely occupied. Unfortunately, experimentally characterizing these conformational ensembles can be costly and inefficient (Terwilliger et al.). Meanwhile, modeling these systems with molecular dynamics (MD) simulation is also expensive, with many conformational changes occurring over millisecond or longer timescales, often exceeding the limits of what can be directly simulated. One recent approach to solve this problem is to use generative models. However, these models aren't well developed for rare protein conformations, and consistent, accurate sampling remains an open problem.

AlphaFold3 (Abramson et al.) solves the protein structure prediction task through diffusion modeling, in which a noised structure follows a learned trajectory to generate a protein structure. However, the sampling distribution of AlphaFold3 is not guaranteed to be physically representative. One of the recent advances in this field is the development of Boltzmann emulators like BioEmu (Lewis et al.). These models generate protein structures similarly to AlphaFold3 but approximate a Boltzmann distribution. While this is more efficient than conducting millisecond-long MD simulations, it is still costly to sample a large number of conformations to obtain ensemble thermodynamic properties. A shortcut to this would be to understand the representations themselves, and obtain these properties with a single prediction from those representations.

Sparse autoencoders (SAEs), self-supervised models which reconstruct the input through a learned latent space, aim to find approximately monosemantic features within pretrained models, enhancing the interpretability of model predictions (Gao et al.). In recent years, their applications have been rapidly expanding. In particular, there have been a variety of ways they have been applied to diffusion models, with one of the main points of contention being the issue of how to incorporate the denoising timestep (Surkov et al.; Kim et al.; Cywiński and Deja; Huang et al.).

In this work, we show that by training SAEs on BioEmu, single features can be extracted that highly correlate with structural and ensemble properties of proteins, improving understanding of these models. In addition, these structural, ensemble, and thermodynamic protein properties can

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

be directly predicted accurately from the hidden embeddings with only a single linear layer on a single forward pass. Previously, obtaining ensemble properties with BioEmu involved extensive sampling, so single linear layer, single pass prediction is a much cheaper alternative. Further, beyond just prediction, the hidden activations of BioEmu can be directly altered to steer towards target protein properties on unseen sequences, as well as bias towards specific conformation states for cryptic pockets, domain motions, and kinases. We show that BioEmu generation can be steered to generate structures with larger or smaller radius of gyration, apo/holo forms of cryptic pockets, domain motion conformers, and DFG-in and DFG-out states in kinases.

Key contributions: We analyze BioEmu's hidden representations, and directly manipulate them to steer towards biophysical ensemble properties and target specific conformers. Our contributions include:

- Training SAEs on BioEmu's hidden activations, extracting single features strongly correlated with biophysical properties, and showing that ensemble and thermodynamic properties can be extracted directly from hidden activations rather than having to conduct extensive, expensive generation.
- Training a small MLP adapter to directly add to the hidden activations of BioEmu at a single denoising timestep and demonstrating that this is enough to semi-quantitatively steer R_{g} of generated structures for unseen proteins.
- Training and benchmarking a similar MLP adapter as the R_{g} adapter at 8 different denoising timesteps spaced throughout the denoising trajectory and showing that this adapter can steer towards specific conformers for cryptic pockets, domain motions, and kinases.

2 Related works

Protein model representation analysis Capturing dynamical protein properties is useful for a multitude of both experimental and computational applications, and even more so when done cheaply (Ebrahimi and Samanta; Watson et al.). To this end, several works have analyzed the representations of structure prediction and protein language models. InterPLM trained SAEs on ESM-2, and showed that interpretable features could be extracted which correspond to structure-level and domain-level features (Simon and Zou). The ESMC work extends the type of analysis done in InterPLM to construct a mapping of a feature to a part of protein space (Candido et al.). AlphaInterp analyzes the information encoded in the single and pair representations of AlphaFold3, evaluating them at different parts of the pairformer process (Feldman and Skolnick). Our work extends this type of analysis to BioEmu, allowing us to analyze and directly predict ensemble features from the hidden activations. Since BioEmu is diffusion-based, we additionally conduct timestep analysis to determine how the representations vary across each denoising timestep, as opposed to ESM-2 and ESMC, which are bidirectional masked transformer architectures instead of diffusion.

Steering protein models In addition to BioEmu, there are other protein model architectures that sample diverse portions of conformation space. AlphaFlow alters AlphaFold2 (Jumper et al.) to use flow matching, allowing for improved conformational diversity compared to MSA subsampling (Jing et al.). ConfDiff trains a SE(3) diffusion model conditioned on ESMFold embeddings with classifier-free guidance, and incorporates a physics-based MD energy function reward to guide generation, improving conformation diversity (Wang et al.). Steering protein models has many precedents in the literature. Diffusion models can be steered at inference time through gradient-based guidance or sequential Monte-Carlo methods (Dhariwal and Nichol; Wu et al.; Bansal et al.). In addition to these, many other works enhance conformational sampling by intervening at the multiple sequence alignment (MSA) stage (Wayment-Steele et al.; Roney and Ovchinnikov; Kalakoti and Wallner; del Alamo et al.; Lee et al., b). More classical enhanced sampling methods include umbrella sampling, metadynamics, and weighted ensemble simulations, which operate through some combination of biasing physical simulations, reweighting, and trajectory resampling (Huber and Kim; Laio and Parrinello; Torrie and Valleau).

Conformation steering Most closely related to our work are BioEmu-CV, ConforNets, and ConforMix. BioEmu-CV learns a small multilayer perceptron (MLP) that alters the single representa-

tions after BioEmu's frozen Evoformer trunk, allowing time-lagged collective variables to be extracted (Park et al.). ConforNets applies channel-wise affine transformations to alter the pair representations to produce active or diverse protein structures (Lee et al., a). ConforMix steers through inference-time twisted diffusion sampling, biasing the reverse diffusion trajectory towards more desired conformations (Richman et al.). Each of these is a different way of steering towards rare conformations, and they are benchmarked on at least two of cryptic pockets, kinases, and domain motions. Our steering shares a similar goal, and is benchmarked on the same set of cryptic pockets, kinases, and domain motions. However, our work accomplishes it by acting directly on the hidden representations rather than modifying the physical potential or resampling trajectories. Further, rather than steering towards rare conformations by encouraging conformation diversity, our work enables steering explicitly towards a desired open/closed state through supervised training. We do this by training on the CryptoBank (Martinez et al.), CryptoBench (Škrhák et al.), Kincore (Modi and Dunbrack), and DynDom (Qi et al.) datasets. This steering is generalizable across unseen proteins and is cross-family, unlike ConforNets, which only generalizes within-family, or sometimes only within-protein.

3 METHODS

3.1 BIOEMU SAMPLING

To obtain a representative sample of proteins, we take 1,000 randomly sampled representative sequences from UniRef50 (The UniProt Consortium). We choose sequences with lengths less than 300 residues and which have InterPro annotations to facilitate later analysis (Blum et al.). We pass these 1000 sequences through BioEmu and capture embeddings from the last invariant point attention (IPA) layer at 10 evenly spaced timesteps throughout the denoising process. These sequences were randomly sampled at training time until all of the sequences were sampled 32 times. Evaluation used a separate set of 1000 sequences, and reference ensemble values were from sequences sampled 128 times. Full analysis details are present in the appendix.

3.2 BIOEMU STEERING

We trained a MLP that alters the hidden activations in BioEmu to steer generation towards a desired biophysical property. The MLP takes the hidden activations and the desired signed standard deviation level away from the property mean as input. In this work, we steer with an SAE feature which correlates strongly with the protein radius of gyration (R_{g}), as such an approach could enable the efficient sampling of conformational states with varying R_{g}. We then extend this to steer towards cryptic pocket opening, kinase DFG-in to DFG-out, and domain motions. The datasets we use are CryptoBank (Martinez et al.), CryptoBench (Škrhák et al.), Kincore (Modi and Dunbrack), and DynDom (Qi et al.).

For cryptic pockets, kinases, and domain motions, instead of injecting at one site, we apply the MLP at 8 timesteps. We filter the training data to exclude the benchmark proteins used by BioEmu, ConforNets, and ConforMix so that we can evaluate and compare to them on the same set of proteins. For cryptic pockets and domain motions, proteins were clustered with MMseqs2 (Steinegger and Söding) using cascaded easy-cluster at a 30% minimum sequence identity and 50% coverage. Easy-search was subsequently run to identify pairs that satisfied both criteria but were placed in different clusters, and affected clusters were merged. The training/validation/test split was then done based on clusters, preventing proteins that are too similar from being split across training, validation, and test sets. We exclude kinases from this filtering as the kinase set is already highly similar. The full implementation details are present in the appendix.

4 EXPERIMENTAL RESULTS

4.1 BIOEMU REPRESENTATION ANALYSIS

BioEmu is aligned such that generated samples approximate a Boltzmann distribution; as such, we hypothesize that the internal representations of the model are able to encode and predict biophysical ensemble properties. To this end, we correlate many different biophysical properties with feature

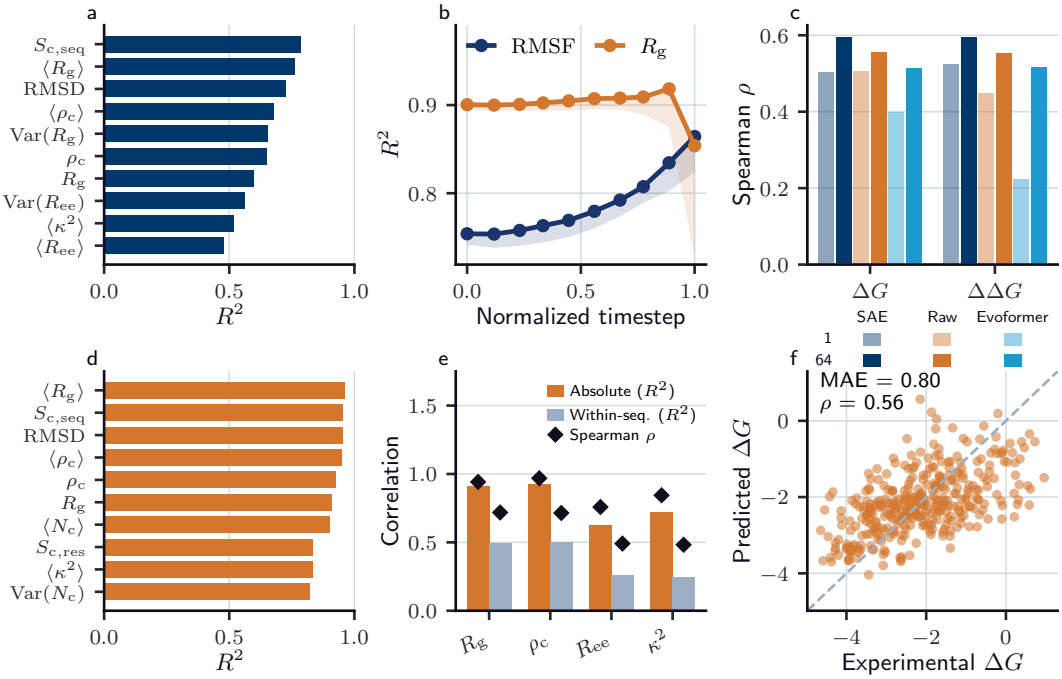

Figure 1: **BioEmu representation analysis.** All reported metrics are calculated with 5-fold cross validation and ridge regression. Property abbreviations are as follows: contact density ρ_c, sequence contact entropy $S_{c,seq}$, residue contact entropy $S_{c,res}$, contact number N_c, relative shape anisotropy κ^2, end-to-end distance R_{ee}, an ensemble mean of a property $\langle\rangle$, an ensemble variance of a property Var(). (a) R^2 values for ridge regression fits on the single best SAE feature for specific properties. (b) RMSF and R_g R^2 values across the denoising timesteps, where 0 corresponds to completely denoised and 1 corresponds to completely noised. The shaded area represents correlation values at different transformer extraction layers. (c) ΔG and $\Delta\Delta G$ folding stability Spearman ρ fits on single best SAE feature and 64 best SAE features. The cross validation set was taken from the test set of the BioEmu folding stability analysis in the original BioEmu paper (Lewis et al.) to prevent any data leakage. (d) R^2 fits on all 512 raw hidden activation units from BioEmu for specific properties. (e) Radius of gyration, contact density, end-to-end distance, and shape anisotropy R^2 and Spearman ρ compared with their relative values. These were fit on the 512 raw hidden activation units, similar to (a) and (b). Relative values are simply regressing on sequence-normalized values rather than the raw values on both axes. (f) Predicted ΔG vs. experimental ΔG for the 64 SAE features case.

activations in our trained SAEs. In (Fig. 1a,d), we depict the top 10 correlated properties for both single SAE features as well as the entire 512 hidden activations with ridge regression.

The correlations for single SAE feature and full raw activations are similar in ordering, with raw activations being much higher. From this, it's apparent that BioEmu does well in predicting ensemble properties. Interestingly, correlation performance drops dramatically between a raw property and its relative (normalized) property (Fig. 1e). The relative property is simply the raw property but with both axes sequence normalized, which largely removes ensemble differences. Thus, much of the strong correlation BioEmu encodes for structural properties like R_g is actually due to its strong ensemble representation rather than single sample representation.

Different features tend to be most strongly represented at different timesteps (Fig. 1b). RMSF increases in correlation at more noised structures, whereas R_g is mostly flat, but with a sharp dip at the very noised structures. Extended timestep analysis is present in the appendix (Fig. A1), but the general trend is that ensemble features have higher correlations at more noised structures, as we see with RMSF, and sample-level structural features tend to have higher correlations at more denoised structures, which is somewhat seen with R_g. This suggests a paradigm where BioEmu encodes ensemble representations earlier in the denoising trajectory, and sample-level representations later in the trajectory.

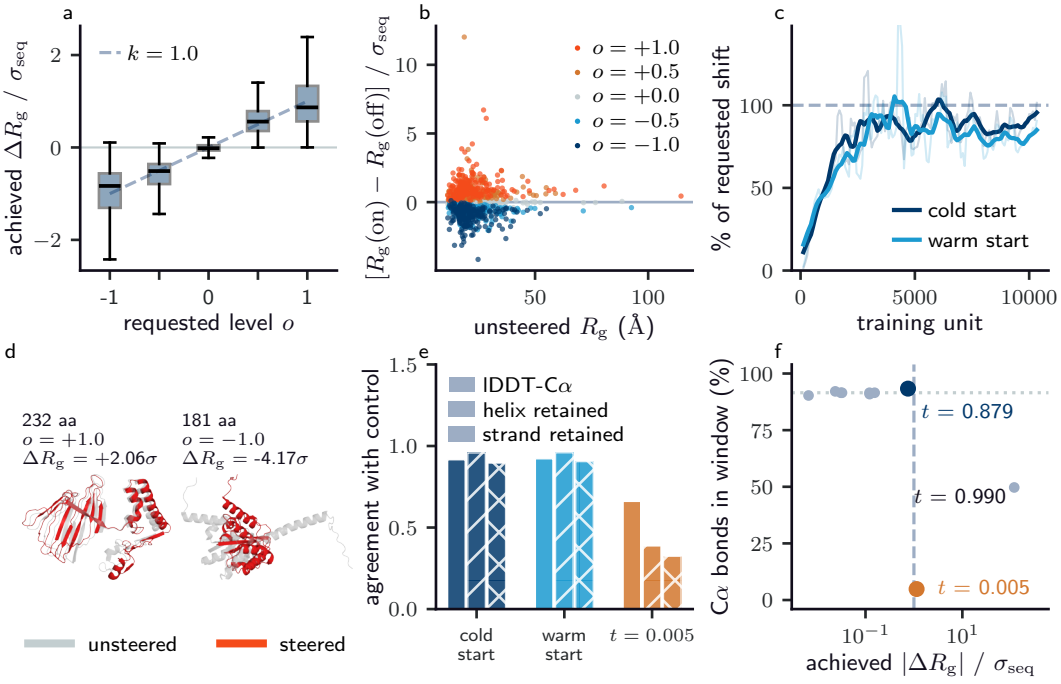

Figure 2: **BioEmu R_g steering.** (a) Box plot of achieved steering vs. requested steering. The dotted line represents optimal steering response. (b) Steering response vs. R_g, colored by target level. (c) Training R_g shift response for cold and warm starts. A training unit is a sample at a specific level. Colored lines are centered averages over 9 low opacity line values, and the low opacity lines represent centered averages of 250 training units. (d) Of 10 randomly sampled structures, the steered and unsteered structures are superimposed for the max, median, and min ΔR_g response. (e) Warm start, cold start, and timestep 0.005 comparison for structure retention. LDDT was calculated based on (Mariani et al.), and helix and strand retention were calculated based on (Labesse et al.). (f) ΔR_g shift after applying a gradient directly from calculated R_g on the hidden activation, capped at 10% of the hidden activation at each score call. The dotted line represents the response necessary to achieve the requested level.

Further, we trained ridge regression probes with a single feature, up to 64 features, on experimental ΔG and $\Delta\Delta G$ values with reference Evoformer priors (Tsuboyama et al.). The SAE latents and raw BioEmu activations performed only marginally better than the Evoformer priors, indicating that BioEmu denoising may be encoding important information for folding stability, but that much of the representations are already encoded in the Evoformer embeddings (Fig. 1c).

Surprisingly, in both ΔG and $\Delta\Delta G$, the ridge regression probes from a single BioEmu sample predicted out-of-fold free energy values at almost the same accuracy as extended sampling in the original BioEmu paper (Lewis et al.). In the BioEmu paper, extended sampling was used to approximate the distribution of structures for a single sequence, and under the assumption of the structures being Boltzmann distributed, calculated ΔG. Here, however, we show that we can instead just sample once per sequence and predict the free energies directly from the hidden representations. In the BioEmu paper, they reported Spearman correlations of 0.6 for both ΔG and $\Delta\Delta G$, which the hidden representations comfortably match at 64 features (Lewis et al.). They also reported MAE values of 0.9 and 0.8 for ΔG and $\Delta\Delta G$, respectively, which the probes match as well (Fig. 1f). Thus, as probes on the hidden representations of BioEmu can predict thermodynamic and ensemble properties with similar accuracy to extended sampling, we conclude that ensemble and thermodynamic information is contained and extractable from BioEmu's hidden representations.

4.2 R_g BioEmu steering

We trained a small MLP adapter that inputs the hidden activations of BioEmu and a steering direction, and outputs a slightly altered hidden dimension which continues through the original BioEmu model. This was trained with reconstruction, contrastive, geometric, and local distance terms and a frozen BioEmu model. For this work, we decided to steer along R_g, with a target input level representing standard deviations away from the ensemble mean of the sequence.

Steering is semi-quantitative, with the mean of the test set sequences following the requested R_g change closely (Fig. 2a). Generation does vary, as unsteered R_g variance is about as large or larger than the requested R_g steering change, but the steering direction is almost always correct (Fig. 2b). Both cold and warm starts, involving biasing the initial direction towards the pure R_g gradient, are able to learn (Fig. 2c). However, the denoising call at which the MLP is inserted matters. Timestep 0.879 is the only point at which the R_g influence is high enough to achieve the target level but not destroy the structure (Fig. 2f). Timesteps 0.990 and 0.005 both have strong effects on R_g, but destroy the structure. Timestep 0.005 in particular is at the very end of the denoising process, so steering at that point effectively rips apart peptide bonds rather than inducing a concerted change across the whole protein. This is further supported in (Fig. 2e), in which secondary structure and strand retention plummet. This suggests that larger-scale motions are encoded at earlier timesteps, and more precise atom placement is encoded at later denoising timesteps. This complements our claim in the first representation analysis section that ensemble-level properties are encoded earlier in the denoising and sample-level properties are encoded later in the denoising. Visualizations of steered proteins are present in (Fig. 2d), showing steered and still intact proteins. These results show that BioEmu's hidden activations can be directly manipulated to semi-quantitatively steer generation towards a desired property, in this case radius of gyration. Further, there is a correct time in the denoising process at which to steer. For radius of gyration, that is very early into the denoising, but may differ for other properties.

4.3 Cryptic pocket, kinase, and domain motion steering

Beyond simply steering towards specific biophysical properties like R_g, we tested whether we could steer towards unseen cryptic pocket apo/holo structures, two different domain motion conformers, and DFG-in and DFG-out kinase states by directly modifying the hidden activations of BioEmu. This is more biologically important, as being able to consistently and cheaply predict these states for unseen proteins would be very helpful for small molecule and drug development. To ensure the train and test splits were sufficiently different, we performed clustering on our full dataset and made the train/validation/test split based on those clusters.

To steer towards these structures, we minimized the aligned RMSD of generated structures, and constrained generation with the same geometry term used in the R_g steering. Similarly to the R_g steering, we initially attempted to steer generation by modifying the activations at a single timestep. Unlike the R_g steering, though, none of the timesteps individually moved the RMSD metric very much. As such, we added the learned MLP adapter to each of the 10 tested denoising timestep spots, with the exception of the very first and very last timestep spots, for a total of 8 spots. This moved the RMSD metric significantly more, indicating that the information to steer towards specific conformers is spread across the denoising process rather than centralized at a single timestep.

To benchmark, we compared against native BioEmu (Lewis et al.), as well as ConforMix (Richman et al.) and ConforNets (Lee et al., a). BioEmu is the pure baseline, as our method acts atop BioEmu. ConforMix steers towards rare conformations through twisted diffusion sampling, which improves sampling of diverse conformations. ConforNets also steers towards rare conformations, but does so by learning affine transformations on the pair representations in the pairformer for each protein.

While native BioEmu is our baseline, benchmarking against ConforMix and ConforNets is a slightly different comparison as ConforMix and ConforNets try to sample more diverse and rare conformations, rather than steer towards a specific conformation, which is what our method does. In particular, ConforMix allows for reweighting to reconstruct the original distribution. Our goal, instead of sampling diverse structures, is to steer towards a specific conformation. For example, given a structure with a cryptic pocket opening, is it possible to steer structures to predominantly sample the apo form of that structure, where we specify in the input that we want the apo form.

Table 1: Steering performance across conformational benchmarks for success@100, with standard error. ConforNets results are the same numbers reported in their paper, denoted with the asterisk, while the rest were recomputed by us.

Benchmark	N	BioEmu	ConforMix	ConforNets*	Ours
Cryptic pockets (apo)	32	0.260 ± 0.077	0.312 ± 0.083	**0.488**	0.250 ± 0.078
Cryptic pockets (holo)	32	0.558 ± 0.082	0.562 ± 0.089	**0.789**	0.625 ± 0.087
Domain motions	44	0.727 ± 0.072	0.727 ± 0.085	**0.819**	0.727 ± 0.072
Kinases (DFG-in)	20	0.800 ± 0.092	**0.850** ± 0.082	–	**0.850** ± 0.082
Kinases (DFG-out)	20	0.610 ± 0.108	0.600 ± 0.112	–	**0.650** ± 0.109

Table 2: Steering performance across conformational benchmarks for success@5, with standard error. ConforNets results are omitted here as they are not reported in the ConforNets paper.

Benchmark	N	BioEmu	ConforMix	Ours
Cryptic pockets (apo)	32	0.161 ± 0.058	**0.246** ± 0.069	0.196 ± 0.067
Cryptic pockets (holo)	32	0.271 ± 0.067	0.378 ± 0.072	**0.434** ± 0.077
Domain motions	44	0.529 ± 0.070	0.485 ± 0.064	**0.537** ± 0.068
Kinases (DFG-in)	20	0.733 ± 0.093	0.653 ± 0.094	**0.739** ± 0.092
Kinases (DFG-out)	20	0.426 ± 0.100	**0.478** ± 0.105	0.472 ± 0.102

We follow similar benchmarking to that done by BioEmu (Lewis et al.) and ConforNets (Lee et al., a), measuring success@B for the same set of structures on cryptic pockets and domain motions, where B is the number of structures sampled to attempt to generate a structure within a specific aligned RMSD τ of the reference structure. Our kinase set differs from ConforNets' slightly as we took sequences with both a DFG-in and a DFG-out structure, which did not apply to all of ConforNets' kinases.

Our steering improves over the BioEmu baseline for most cryptic pockets and kinases, and outperforms ConforMix on a decent number of metrics as well. The ConforNets values are more competitive, with their values outperforming ours. Significantly, though, for this benchmark, ConforNets was trained on each protein individually, while ours generalizes to unseen proteins. ConforNets also contains a within-family transfer intervention, which applies to kinases, but does not apply to cryptic pockets or domain motions.

Although our method shows enhanced performance compared to the BioEmu baseline for cryptic pockets and kinases, the same is not entirely true for domain motion conformers. In fact, our method is comparable, but not better in the success@B metric. This is likely because domain motions aren't a very well-defined set of conformers, with the benchmark and the training set not having very similar conformer definitions. This results in the trained adapter trying to steer towards conformers that aren't defined as correct in the benchmark. As such, one flaw with our method is that the target conformation has to be well defined, and there must be available training data. Still, while we do not perform as well on the success@100 metric, when taking the mean of all successes rather than only the max per 100 samples, our domain motion training is actually quite competitive (Fig. 3f). So, while the domain motion case isn't perfectly well defined, our method does well where it is.

In (Fig. 3a, b, c) the success@B performance of the different methods is compared as B varies. On average, our method performs comparatively better at lower B than at higher B. Due to our method biasing all structures towards a specific conformation, our success increases rapidly at low B, but saturates more quickly as a result. This is also pertinently shown in (Fig. 3f), which measures the fraction of successful samples over all samples taken. On average, our method reaches the targeted conformation more often than the other methods, but this effect is diluted with the standard success@B metric, which only takes the single best sample. As such, our method is better for targeting a single rare conformation state, and doing so with few samples. Although diversification methods like ConforMix and ConforNets improve overall rare conformation sampling, our method allows targeted sampling with fewer generations.

All timesteps seem to play a large role in steering. In (Fig. 3d), the residual fraction at all timesteps for all cases is above 0.5, indicating that there is a large intervention at each point. The exception

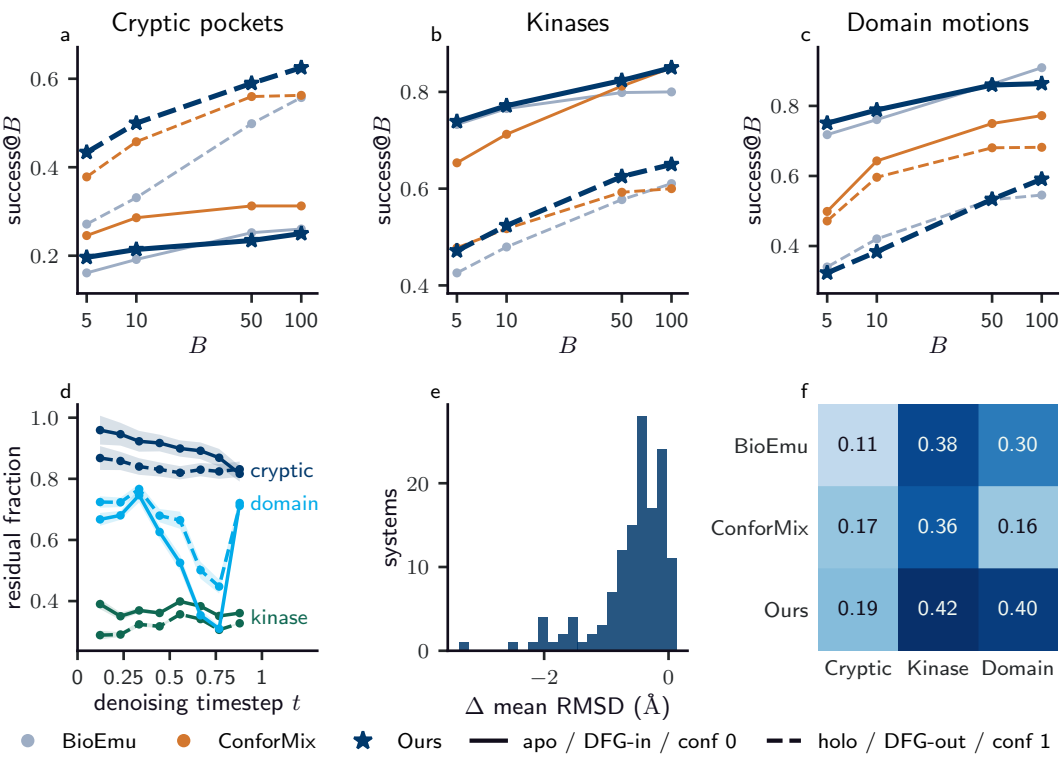

Figure 3: **BioEmu conformation steering.** In the domain motion benchmark, we defined conformer 0 to be the conformer more often sampled by BioEmu, and conformer 1 the one less often sampled by BioEmu, allowing our training and metrics to be defined for domain motion. (a) Success@B vs. B for cryptic pocket steering, comparing BioEmu, ConforMix, and our method. (b) Success@B vs. B for kinase steering, comparing BioEmu, ConforMix, and our method. (c) Success@B vs. B for domain motion steering, comparing BioEmu, ConforMix, and our method. (d) Average magnitude of the residual fraction added at each denoising timestep. (e) Change in mean RMSD for each steered draw in our own test set, which contains a larger batch of proteins than the benchmark, compared to the unsteered structure for holo cryptic pocket structures. (f) Fraction of successfully steered structures, using the total number successfully steered out of the 100 rather than only taking the best structure out of the 100. The same τ as the one in the benchmark is used.

is the domain motion case, which features a sharp dip around $t = 0.75$. Besides this, though, the relative fractions at the other timesteps don't vary much. This is in contrast to the R_g steering, which was able to be steered at a single timestep, and limited to a single timestep fairly early in the denoising process ($t = 0.879$). In the previous two sections we observed that ensemble-level properties and large-scale motions tend to be encoded earlier in the denoising, and sample-level and specific atom-level placement tend to be encoded later in the denoising process. Cryptic pocket apo/holo states, kinase DFG-in and DFG-out states, and domain motion conformers all require both large and small adjustments, so all timesteps affecting steering support this argument.

In addition to our improvement on the success@B metric on the benchmark, we also improve on our own test set. In (Fig. 3e), we depict the Δ mean RMSD improvement over 25 draws per sequence per conformer for the holo cryptic pocket conformer. In almost every single test sequence, our method moves the sample closer to the targeted conformer, showing that our method is able to identify the correct direction to move and generalizes to proteins it has never seen before. Test set Δ RMSD graphs for the other cases are present in the appendix (Fig. A7).

5 DISCUSSION

We began by investigating what is encoded in BioEmu's hidden representations, learning that biophysical properties and importantly ensemble and thermodynamic properties are encoded directly in the hidden activations. It is possible to extract these ensemble properties with a single ridge regression layer and achieve comparable accuracy to extended sampling of BioEmu, a result that makes these properties cheaper to obtain. Significantly, a trained ridge regression probe is generalizable, making these properties obtainable for a vast number of proteins. With the knowledge that the hidden representations encode these properties, we then analyzed how the representations of these properties change across the denoising trajectory. From this, we learned that ensemble-level properties tend to be encoded earlier in the denoising, and sample-level properties later into the denoising.

Beyond prediction, we investigated if we could manipulate the hidden activations directly to influence a targeted biophysical property in the generated protein structure. For this, we chose R_{g}, and learned that R_{g} can be semi-quantitatively steered through the hidden activations. Importantly, this can be done at a single timestep, indicating that a sufficient amount of R_{g} information is encoded at a single point in BioEmu's hidden representations to allow for successful steering. The timestep that worked for this was $t = 0.879$, relatively early into the denoising process, suggesting that larger movements are encoded earlier in the denoising process. This was further supported by tests at $t = 0.005$, which did steer R_{g}, but did so by breaking peptide bonds, likely because only smaller scale atom placement is encoded at the very late denoising steps.

Beyond biophysical properties, we then investigated if we could directly target specific conformers in cryptic pockets, domain motions, and kinases. We were unable to do so at a single timestep, but were able to by intervening at 8 different timesteps spaced throughout the denoising process. Importantly, our method is generalizable outside of the training set, and acts only through the hidden activations. It is also different from the other methods we compared to, ConforMix (Richman et al.) and ConforNets (Lee et al., a), as their methods diversify sampling, whereas ours targets a specific conformer, and if anything compresses towards the conformer rather than diversifying. This is important, as it allows our method to cheaply generate target conformers for unseen cryptic pockets, domain motions, and kinases. In addition, there are many other protein classes and conformers not tested in this work, but that likely can be steered towards and made cheap to generate through direct manipulation of hidden activations.

6 CONCLUSION

This work evaluates what properties are encoded in the hidden activations of BioEmu, and how structure generation can be steered by manipulating these hidden activations. Through correlation and direct prediction, we found that BioEmu strongly encodes ensemble and thermodynamic properties, with a weaker sample-specific representation. Further, in addition to just activation correlation, BioEmu activations can predict ensemble and thermodynamic properties to a similar degree as extended sampling, which demonstrates that information sufficient to predict these quantities is available from a single BioEmu trajectory. Beyond simply encoding, we also show that structural properties such as R_{g} can be directly steered during generation to semi-quantitatively influence the final structure if injected at an appropriate timestep. Finally, in addition to pure biophysical properties, BioEmu activations can be directly manipulated to steer generation of unseen proteins toward cryptic pocket apo/holo states, domain motion conformers, and kinase DFG-in and DFG-out states.

Limitations Training and evaluation of the SAEs were only conducted with 1000 sequences of fewer than 300 residues. Although all reported metrics are out-of-fold, a larger and more diverse set of sequences would improve generalizability. The steering mechanism has only been tested with R_{g}, and only with BioEmu. Cryptic pocket, domain motion, and kinase steering was evaluated on a low number of systems, and results may differ when evaluated on a larger number of systems. Additional work could aim to test other properties, including ensemble properties, test with more sequences, test with larger model interventions, and test with other protein models.

AI USE STATEMENT

In this work, we used generative AI tools for proposing and refining hypotheses, design and feedback on research methodology, implementing methods, and cleaning datasets. We have not used generative AI tools for supporting qualitative and thematic data analysis or interpreting results. Generating synthetic datasets, developing theoretical models, formulating mathematical claims, proving mathematical claims, assisting in writing of proofs, and assisting with translation are not applicable to this work. Additionally, we used generative AI tools for creating scientific figures, writing code, summarizing existing literature, brainstorming, and identifying relevant literature. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature search, and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

RESPONSIBLE USE STATEMENT

Our models are designed as interpretability tools and for steering generation. They could potentially be misused in applications like drug design, and should not be taken as experimentally validated structures. We recommend predictions be used with appropriate experimental validation and established biosafety practices.

REPRODUCIBILITY STATEMENT

All code, data, and models can be found at: https://anonymous.4open.science/r/confsteer_bioemu-43B3

REFERENCES

J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. 630(8016):493–500. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://www.nature.com/articles/s41586-024-07487-w.

A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. URL http://arxiv.org/abs/2302.07121.

M. Blum, A. Andreeva, L. Florentino, S. Chuguransky, T. Grego, E. Hobbs, B. Pinto, A. Orr, T. Paysan-Lafosse, I. Ponamareva, G. Salazar, N. Bordin, P. Bork, A. Bridge, L. Colwell, J. Gough, D. Haft, I. Letunic, F. Llinares-López, A. Marchler-Bauer, L. Meng-Papaxanthos, H. Mi, D. Natale, C. Orengo, A. Pandurangan, D. Piovesan, C. Rivoire, C. A. Sigrist, N. Thanki, F. Thibaud-Nissen, P. Thomas, S. E. Tosatto, C. Wu, and A. Bateman. InterPro: the protein sequence classification resource in 2025. 53:D444–D456. ISSN 1362-4962. doi: 10.1093/nar/gkae1082. URL https://doi.org/10.1093/nar/gkae1082.

S. Candido, T. Hayes, A. Derry, R. Rao, Z. Lin, R. Verkuil, B. Z. Wu, J. S. Lee, E. S. Bruguera, J. A. Keval, M. Kopylov, J. E. Pak, W. Wu, N. Thomas, S. Mataraso, A. Hsu, A. C. Trotman-Grant, K. Fatras, A. d. S. Costa, R. Badkundri, H. Akın, D. Oktay, J. Deaton, E. Montabana, H. Sitwala, Y. Yu, M. Wiggert, D. A. Carlin, A. W. Goering, T. Blazejewski, M. Sandora, M. Hla, T. Z. Jia, L. H. Kloker, N. J. Sofroniew, M. Uehara, J. Pannu, S. Bachas, D. S. Liu, T. Sercu, and A. Rives. Language modeling materializes a world model of protein biology. URL https://www.biorxiv.org/content/10.64898/2026.06.03.729735v1. ISSN: 2692-8205 Pages: 2026.06.03.729735 Section: New Results.

B. Cywiński and K. Deja. SAeUron: Interpretable concept unlearning in diffusion models with sparse autoencoders. URL http://arxiv.org/abs/2501.18052.

D. del Alamo, D. Sala, H. S. Mchaourab, and J. Meiler. Sampling alternative conformational states of transporters and receptors with AlphaFold2. 11:e75751. ISSN 2050-084X. doi: 10.7554/eLife.75751. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC9023059/.

P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. URL http://arxiv.org/abs/2105.05233.

S. B. Ebrahimi and D. Samanta. Engineering protein-based therapeutics through structural and chemical design. 14(1):2411. ISSN 2041-1723. doi: 10.1038/s41467-023-38039-x. URL https://www.nature.com/articles/s41467-023-38039-x.

J. Feldman and J. Skolnick. AlphaInterp: Mechanistic interpretability of AlphaFold 3 reveals how evolutionary information shapes protein structure prediction. URL https://www.biorxiv.org/content/10.64898/2026.04.22.720175v2. ISSN: 2692-8205 Pages: 2026.04.22.720175 Section: New Results.

L. Gao, T. D. l. Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. URL http://arxiv.org/abs/2406.04093.

V. S.-J. Huang, L. Zhuo, Y. Xin, Z. Wang, F.-Y. Wang, Y. Wang, R. Zhang, P. Gao, and H. Li. TIDE : Temporal-aware sparse autoencoders for interpretable diffusion transformers in image generation. URL http://arxiv.org/abs/2503.07050.

G. A. Huber and S. Kim. Weighted-ensemble brownian dynamics simulations for protein association reactions. 70(1):97–110. ISSN 0006-3495. doi: 10.1016/S0006-3495(96)79552-8. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC1224912/.

B. Jing, B. Berger, and T. Jaakkola. AlphaFold meets flow matching for generating protein ensembles. URL http://arxiv.org/abs/2402.04845.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. 596(7873):583–589. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Y. Kalakoti and B. Wallner. AFsample2 predicts multiple conformations and ensembles with AlphaFold2. 8(1):373. ISSN 2399-3642. doi: 10.1038/s42003-025-07791-9. URL https://www.nature.com/articles/s42003-025-07791-9.

D. Kim, X. Thomas, and D. Ghadiyaram. $\textit{Revelio}$: Interpreting and leveraging semantic information in diffusion models. URL http://arxiv.org/abs/2411.16725.

G. Labesse, N. Colloc'h, J. Pothier, and J.-P. Mornon. P-SEA: a new efficient assignment of secondary structure from c trace of proteins. 13(3):291–295. ISSN 1367-4803. doi: 10.1093/bioinformatics/13.3.291. URL https://doi.org/10.1093/bioinformatics/13.3.291.

A. Laio and M. Parrinello. Escaping free-energy minima. 99(20):12562–12566. ISSN 0027-8424. doi: 10.1073/pnas.202427399. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC130499/.

M. Lee, C. Kalicki, M. Jeon, A. Qabel, A. Fadini, and M. AlQuraishi. ConforNets: Latents-based conformational control in OpenFold3, a. URL http://arxiv.org/abs/2604.18559.

M. Lee, J. W. Schafer, J. Prabakaran, D. Chakravarty, M. F. Clore, and L. L. Porter. Large-scale predictions of alternative protein conformations by AlphaFold2-based sequence association. 16(1):5622, b. ISSN 2041-1723. doi: 10.1038/s41467-025-60759-5. URL https://www.nature.com/articles/s41467-025-60759-5.

S. Lewis, T. Hempel, J. Jiménez Luna, M. Gastegger, Y. Xie, A. Y. K. Foong, V. García Satorras, O. Abdin, B. S. Veeling, I. Zaporozhets, Y. Chen, S. Yang, A. Schneuing, J. Nigam, F. Barbero, V. Stimper, A. Campbell, J. Yim, M. Lienen, Y. Shi, S. Zheng, H. Schulz, U. Munir, R. Tomioka, C. Clementi, and F. Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. URL http://biorxiv.org/lookup/doi/10.1101/2024.12.05.626885.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. URL http://arxiv.org/abs/1711.05101.

V. Mariani, M. Biasini, A. Barbato, and T. Schwede. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. 29(21):2722–2728. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt473. URL https://doi.org/10.1093/bioinformatics/btt473.

P. F. Martinez, T. Fröhlking, A. Borsatto, and F. L. Gervasio. CryptoBank: A resource for the identification and prediction of cryptic sites in proteins. 12(17):eady6364. ISSN 2375-2548. doi: 10.1126/sciadv.ady6364. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC13267282/.

M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger. ColabFold: making protein folding accessible to all. 19(6):679–682. ISSN 1548-7105. doi: 10.1038/s41592-022-01488-1. URL https://www.nature.com/articles/s41592-022-01488-1.

V. Modi and R. Dunbrack. Kincore: a web resource for structural classification of protein kinases and their inhibitors. URL https://www.biorxiv.org/content/10.1101/2021.02.12.430923v1. Pages: 2021.02.12.430923 Section: New Results.

S. Park, K. Seong, S. Yang, R. Gómez-Bombarelli, and S. Ahn. Learning collective variables from BioEmu with time-lagged generation. URL http://arxiv.org/abs/2507.07390.

G. Qi, R. Lee, and S. Hayward. A comprehensive and non-redundant database of protein domain movements. 21(12):2832–2838. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/bti420. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti420.

D. D. Richman, J. Karaguesian, C.-M. Suomivuori, and R. O. Dror. Unlocking hidden biomolecular conformational landscapes in diffusion models at inference time. URL http://arxiv.org/abs/2512.03312.

J. P. Roney and S. Ovchinnikov. State-of-the-art estimation of protein model accuracy using AlphaFold. 129(23):238101. doi: 10.1103/PhysRevLett.129.238101. URL https://link.aps.org/doi/10.1103/PhysRevLett.129.238101.

E. Simon and J. Zou. InterPLM: discovering interpretable features in protein language models via sparse autoencoders. 22(10):2107–2117. ISSN 1548-7105. doi: 10.1038/s41592-025-02836-7. URL https://www.nature.com/articles/s41592-025-02836-7.

M. Steinegger and J. Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. 35(11):1026–1028. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://www.nature.com/articles/nbt.3988.

V. Surkov, C. Wendler, A. Mari, M. Terekhov, J. Deschenaux, R. West, C. Gulcehre, and D. Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. URL http://arxiv.org/abs/2410.22366.

T. C. Terwilliger, D. Stuart, and S. Yokoyama. Lessons from structural genomics. 38:371–383. ISSN 1936-122X. doi: 10.1146/annurev.biophys.050708.133740. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC2847842/.

The UniProt Consortium. UniProt: the universal protein knowledgebase. 45:D158–D169. ISSN 0305-1048. doi: 10.1093/nar/gkw1099. URL https://doi.org/10.1093/nar/gkw1099.

G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. 23(2):187–199. ISSN 0021-9991. doi: 10.1016/0021-9991(77)90121-8. URL https://www.sciencedirect.com/science/article/pii/0021999177901218.

K. Tsuboyama, J. Dauparas, J. Chen, E. Laine, Y. Mohseni Behbahani, J. J. Weinstein, N. M. Mangan, S. Ovchinnikov, and G. J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. 620(7973):434–444. ISSN 1476-4687. doi: 10.1038/s41586-023-06328-6. URL https://www.nature.com/articles/s41586-023-06328-6.

Y. Wang, L. Wang, Y. Shen, Y. Wang, H. Yuan, Y. Wu, and Q. Gu. Protein conformation generation via force-guided SE(3) diffusion models. URL http://arxiv.org/abs/2403.14088.

J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De novo design of protein structure and function with RFdiffusion. 620(7976):1089–1100. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL https://www.nature.com/articles/s41586-023-06415-8.

H. K. Wayment-Steele, A. Ojoawo, R. Otten, J. M. Apitz, W. Pitsawong, M. Hömberger, S. Ovchinnikov, L. Colwell, and D. Kern. Predicting multiple conformations via sequence clustering and AlphaFold2. 625(7996):832–839. ISSN 1476-4687. doi: 10.1038/s41586-023-06832-9. URL https://www.nature.com/articles/s41586-023-06832-9.

L. Wu, B. L. Trippe, C. A. Naesseth, D. M. Blei, and J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. URL http://arxiv.org/abs/2306.17775.

V. Škrhák, M. Novotný, C. P. Feidakis, R. Krivák, and D. Hoksza. CryptoBench: cryptic protein–ligand binding sites dataset and benchmark. 41(1):btae745. ISSN 1367-4803. doi: 10.1093/bioinformatics/btae745. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC11725321/.

A METHOD DETAILS

A.1 SAE ARCHITECTURE

The SAE architecture used is modeled after the one used in Scaling and evaluating sparse autoencoders, with the addition of a time-aware learned MLP, which was modeled after the one used in TIDE (Gao et al.; Huang et al.). As such, ReLU and TopK activations were used, as well as an auxiliary K of 256 for auxiliary reconstruction. The auxiliary coefficient was set to 0.03125 and the dead feature threshold to one million tokens. A latent dimension of 4096 (expansion factor of 8) and topk of 32 were used. For training, the learning rate was set to 1e-4 with a linear warmup of 300 steps starting from 3e-5 and linear decay back to 3e-5. This was trained for 1 epoch, which turned out to be 7098 optimizer steps with a batch size of 8,192. This batch was sampled from a larger batch of 32,768 to ensure a single batch didn't contain all the examples of a given sequence. AdamW was used for the optimizer (Loshchilov and Hutter).

A.2 BioEmu representation analysis

Unless otherwise specified, all analysis was conducted with a separate set of 1000 evaluation sequences. Two main areas were analyzed: sequence-level, and structure-level. For sequence-level, InterPro annotations were used to calculate domain-level F1 scores for each trained SAE feature, as used in InterPLM (Blum et al.; Simon and Zou). For structure-level, RMSF, R_g (C_α only), and contact density were evaluated for each sequence and sample based on Spearman and Pearson correlations with 5-fold cross validation. Contact density used a distance cutoff of 8Å, minimum sequence separation of 3 residues, and referenced to the structure generated with seed 0.

Relative properties (like relative R_g) were calculated by z-score normalizing within sequence. Displayed feature activations are mean pooled within sequence first, then mean pooled across sequences to prevent longer sequences from dominating.

The Delta G probes were trained and evaluated with the closed-form solution of ridge regression, similar to the RMSF probe. Each point used in the resulting correlation analysis was the average of 4 samples of each sequence to reduce variability. For baselines to compare the probe performance to, embeddings were taken directly from the Evoformer output for BioEmu. Pairwise representations from the Evoformer for any given residue were averaged across all of their pairs to give the correct dimensionality input. The folding stability data used came from the test set of the original BioEmu paper, preventing data leakage of pretraining data into our own test set.

A.3 Extended SAE implementation details

BioEmu was run with the DPM denoiser. Evoformer embeddings were calculated once per sequence, cached, and reaccessed for each sample. BioEmu embeddings are of dimension 512, which is what is input and reconstructed by the SAEs. Embeddings are taken from 10 evenly spaced points on the denoising trajectory, with reported normalized timestep 0 corresponding to most denoised, and 1 to most noised.

The SAEs are input normalized across the feature axis, and restored during decoding. The timestep shift is defined as follows:

$$\hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}} \cdot (1 + \text{scale}) + \text{shift}$$

where scale and shift are MLPs that input timestep, expand to 128 hidden dimension with a GELU activation, then output to the dimension of $\boldsymbol{x}$ (Huang et al.). For the SAEs, encoder weights are initialized on a Kaiming uniform distribution, decoder weights are initialized as the transpose of the encoder weights, and biases are zero-initialized. Loss for the SAE is the standard MSE with auxiliary k term (Gao et al.).

All evaluated properties: R_g, contact density, end-to-end distance, shape anisotropy, rmsf, contact number, local q, pair contact entropy, contact probability, pairwise RMSD, and all of the means and variances of these properties when applicable. All evaluations were performed and reported with 5-fold cross validation.

RMSF and other probes evaluate per sample, use ridge regression, and solve for the closed-form solution. 5-fold cross validation along sequence is implemented. For baselines to compare the probe performance to, embeddings were taken directly from the Evoformer output for BioEmu. Pairwise representations from the Evoformer for any given residue were averaged across all of their pairs to give the correct dimensionality input.

A.4 SAE Ablations

Five ablations were conducted. One ablation removed the time dependent MLP, one ablation altered the topk from 32 to 64, and the other three ablations took activations from the 3rd to last, 5th to last, and 7th to last layers of IPA rather than the last.

A.5 R_g STEERING DETAILS

Output weights and biases are initialized to zero to prevent initially biasing structure generation, and hidden weights and biases are initialized to Kaiming uniform. To encode the correct direction, as the signal is initially smaller than the noise and the contrastive term uses output gradients initialized to 0, a warm start is used. The pure R_g gradient rather than the loss gradient is calculated from an unsteered structure, and then applied to $\boldsymbol{W}_{\text{out}}$ as a regression objective to obtain a new initialization for $\boldsymbol{W}_{\text{out}}$ that is in the direction of R_g variation. Training values are averaged across 4 samples per sequence, with a batch size of 5 and levels of [-1, -0.5, 0, 0.5, 1]. Reference sequence mean and standard deviation values are obtained from 32 BioEmu samples. The AdamW optimizer is used.

The MLP follows the following dimensionality: $1024/1 \to 64/64 \to 64 \to 512$, where the / represents separate branches, concatenated at the $64/64$ layer. The input 1024 is comprised of $[\boldsymbol{h}, \bar{\boldsymbol{h}}]$, and the 1 is the target level. The original hidden activation is represented by $\boldsymbol{h}$, and the altered activation for steered runs becomes $\boldsymbol{h}' = \boldsymbol{h} + \Delta\boldsymbol{h}$. More concretely:

$$
\begin{aligned}
\boldsymbol{h}_{1,1} &= \text{LayerNorm}\left(\text{GELU}\left(\text{Linear}([\boldsymbol{h}, \bar{\boldsymbol{h}}], 64)\right)\right), \\
\boldsymbol{h}_{1,2} &= \text{Linear}(o, 64), \\
\boldsymbol{h}_2 &= \text{GELU}\left(\text{Linear}([\boldsymbol{h}_{1,1}, \boldsymbol{h}_{1,2}], 64)\right), \\
\Delta\boldsymbol{h} &= \text{Linear}(\boldsymbol{h}_2, 512).
\end{aligned}
$$

Reconstruction penalizes a steered structure not outputting the targeted R_g. Contrastive loss mitigates structure damage, measured by the antisymmetric term (positive steering - negative steering), as structures tended to increase in R_g more than they should relative to the opposite steering direction. Geometric and local distance terms help prevent unphysical structures and preserve structure generation in BioEmu's original posterior distribution.

Loss objective, including reconstruction, contrastive, regulatory, geometric, and lddt:

$$
\mathcal{L} = \lambda_y \mathcal{L}_y + \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r + \lambda_g \mathcal{L}_g + \lambda_l \mathcal{L}_l
$$

$$
\begin{aligned}
\mathcal{L}_y &= \frac{1}{N} \sum_n \left(\frac{\hat{R}_n - t_n}{\sigma_{s(n)}} \right)^2, \\
\mathcal{L}_c &= \frac{1}{|\mathbb{P}|} \sum_{(p,q)\in\mathbb{P}} \left(\frac{(\hat{R}_p - \hat{R}_q) - (t_p - t_q)}{\sigma_{s(p)}} \right)^2, \\
\mathcal{L}_r &= \frac{1}{NM} \sum_{n,m} \left(\frac{F_{\Delta,(n,m)}}{\rho_{\max}} \right)^2, \\
\mathcal{L}_g &= \frac{1}{N} \sum_n \left(V_n^{\text{win}} + V_n^{\text{clash}} \right), \\
\mathcal{L}_l &= 1 - \frac{1}{P_0 N_l} \sum_{\substack{i<j,\ j-i\le W \\ m_i = m_j = 1}} s_{ij}.
\end{aligned}
$$

where $\hat{R}_n$ is the target property calculated from the generated structure, σ_s is the standard deviation of the target property for the input sequence s, $\mathbb{P}$ is the set of all level pair combinations, $A_{\boldsymbol{\theta}}$ is the function represented by the learned MLP, o_n is the level being trained at, k is a hyperparameter to control property standard deviation response, N is the batch size, M is the number of samples per sequence, and $\rho_{\max}$ is the maximum deviation of $\Delta\boldsymbol{h}$ from $\boldsymbol{h}$, which we set to 0.1.

For $\mathcal{L}_y$, $\mathcal{L}_c$, and $\mathcal{L}_r$:

$$
t_n = \mu_{s(n)} + o_n k \sigma_{s(n)},
$$

$$\bar{\boldsymbol{h}} = \frac{\sum_i m_i \boldsymbol{h}_i}{\sum_i m_i}, \qquad \boldsymbol{r}_i = A_{\boldsymbol{\theta}}\big([\boldsymbol{h}_i, \bar{\boldsymbol{h}}, o]\big),$$

$$\mathrm{RMS}(\boldsymbol{x}) = \sqrt{\frac{\sum_i m_i \|\boldsymbol{x}_i\|^2}{d_h \sum_i m_i}}, \qquad a = \mathrm{RMS}(\boldsymbol{h}), \quad u = \mathrm{RMS}(\boldsymbol{r}),$$

$$F_{\mathrm{raw}} = \frac{u}{a}, \qquad c = \rho_{\max} a,$$

$$\Delta \boldsymbol{h}_i = \boldsymbol{r}_i \beta(u,c), \qquad \beta(u,c) = \begin{cases} 0, & c = 0, \\ 1, & u = 0 \text{ and } c > 0, \\ \dfrac{c \tanh(u/c)}{u}, & u > 0 \text{ and } c > 0. \end{cases}$$

$$F_\Delta = \rho_{\max} \tanh\left(\frac{F_{\mathrm{raw}}}{\rho_{\max}}\right)$$

where $d_h = 512$ is the hidden representation dimensionality.

For $\mathcal{L}_g$, we use a geometric loss penalty similar to AlphaFold2 (Jumper et al.):

$\boldsymbol{x}_{n,i} \in R^3$ is the C_α coordinate of residue i in batch member n, and $m_{n,i} \in \{0,1\}$ is the validity mask. Over residue offsets of 1 with distance $d_{n,i} = \|\boldsymbol{x}_{n,i+1} - \boldsymbol{x}_{n,i}\|_2$, $[\ell, u] = [3.6, 4.0]$ and pair validity $p_{n,i} = m_{n,i} m_{n,i+1}$:

$$h_{[\ell,u]}(d) = \max(d - u, 0) + \max(\ell - d, 0)$$

$$V_n^{\mathrm{win}} = \frac{\sum_i p_{n,i}\, h_{[\ell,u]}(d_{n,i})^2}{\max\left(1,\ \sum_i p_{n,i}\right)}$$

Non-adjacent pairs only, $\Delta = 3$, clash floor $c_{\mathrm{clash}} = 4.1$ Å, with $q_{n,ij} = m_{n,i} m_{n,j}\, \mathbf{1}_{|i-j|>\Delta}$:

$$V_n^{\mathrm{clash}} = \frac{\sum_{i,j} q_{n,ij}\ \max(c_{\mathrm{clash}} - \|\boldsymbol{x}_{n,i} - \boldsymbol{x}_{n,j}\|_2, 0)^2}{\max\left(1,\ \sum_{i,j} q_{n,ij}\right)}$$

For $\mathcal{L}_l$, we use a LDDT loss modeled after the one in AF3 (Abramson et al.), where $\boldsymbol{x}$ is the steered C_α coordinates, $\tilde{\boldsymbol{x}}$ is the unsteered C_α coordinates from the same noise seed, $\mathbb{T}$ is the set of LDDT thresholds from the original LDDT paper (Mariani et al.), and W is the max sequence separation:

$$\delta_{ij} = \big|\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 - \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2\big|$$

$$s_{ij} = \frac{1}{|\mathbb{T}|} \sum_{\tau \in \mathbb{T}} \sigma(\tau - \delta_{ij}), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$P_0 = \frac{1}{|\mathbb{T}|} \sum_{\tau \in \mathbb{T}} \sigma(\tau) \approx 0.80408\ldots$$

$$N_l = \#\left\{(i,j) \mid i < j,\ j - i \le W,\ m_i = m_j = 1\right\}$$

$$\mathbb{T} = \{0.5, 1, 2, 4\}, \qquad W = 8$$

Sequence separation is used instead of spatial distance for pair selection to prevent working against the R_g objective. Loss parameters used for the base case, shown in results: $\lambda_y = 0.01, \lambda_c = 1, \lambda_g = 0.01, \lambda_l = 0, \lambda_r = 0$.

A.6 R_g STEERING ABLATIONS

In addition to the base architecture, seven other architecture tweaks were tested. They are as follows: cold start, omitting the warm start described in the extended steering details; inserting the MLP at timestep 0.005 rather than 0.879; excluding the geometric loss term; including the lddt term at 0.1 λ_l; inserting the level directly into the input alongside the hidden activations, turning the input layer from 1024 to 1025; averaging gradient steps over 4 sequences instead of 1; and omitting the contrastive term.

A.7 CRYPTIC POCKET, DOMAIN MOTION, AND KINASE STEERING OBJECTIVES

In contrast to R_g steering which alters the hidden activations at one site, cryptic pocket, domain motion, and kinase steering alter the hidden activations at 8 different timesteps. The denoiser timesteps we applied the MLP at are: 0.8794, 0.7680, 0.6671, 0.5561, 0.4450, 0.3339, 0.2329, and 0.1221. These are the 8 interior timesteps among 10 evenly spaced denoiser calls, excluding the very first and last. Rather than using a level input, we trained separate adapters for each apo, holo, conformer 1, conformer 2, DFG-in, and DFG-out case. The MLP adapter is exactly the same as in the R_g steering, but the level is not concatenated into the $\boldsymbol{h}_2$ expression.

Cryptic-pocket steering. Proteins came from CryptoBank (Martinez et al.) and CryptoBench (Škrhák et al.). We discarded duplicate sequences, sequences without injective apo to holo mapping, sequences with over 500 residues, sequences where residue maps are inconsistent, and sequences used in the community cryptic pocket benchmark. We then further filtered on RMSD of the N/CA/C/O atoms of pocket (defined in the dataset) and context residues (residues within $12\mathring{A}$ of a pocket residue) being greater than $1.5\mathring{A}$. This resulted in 693 usable proteins. Then, proteins were clustered with MMseqs2 (Steinegger and Söding) using cascaded easy-cluster at a 30% minimum sequence identity and 50% coverage. Easy-search was subsequently run to identify pairs that satisfied both criteria but were placed in different clusters, and affected clusters were merged. The training/validation/test split was then done based on clusters.

For cryptic-pocket steering, the total objective is

$$\mathcal{L}_{\text{cryptic}} = \lambda_m \mathcal{L}_m + \lambda_g \mathcal{L}_g$$

The direct structural target loss is

$$\mathcal{L}_m = \frac{1}{B} \sum_{b=1}^{B} \left(\frac{m_b}{s_b} \right)^2,$$

with

$$m_b = \text{RMSD}_{\text{align}} \left(\boldsymbol{X}_b, \boldsymbol{Y}_{r_b,b}; \mathbb{A}_b \right), \qquad r_b \in \{\text{apo}, \text{holo}\},$$

where $\mathbb{A}_b$ denotes the set of atoms (N, CA, C, O atoms on residues in the cryptic pocket-context range) belonging to the cryptic-pocket metric residue ranges. Pocket residues are defined by the dataset, and context residues are defined as residues within $12\mathring{A}$ of pocket residues. The corresponding apo–holo reference separation is

$$s_b = \text{RMSD}_{\text{align}} \left(\boldsymbol{Y}_{\text{apo},b}, \boldsymbol{Y}_{\text{holo},b}; \mathbb{A}_b \right).$$

Aligned RMSD is defined as

$$\text{RMSD}_{\text{align}}(\boldsymbol{X}, \boldsymbol{Y}; \mathbb{A}) = \left[\frac{1}{|\mathbb{A}|} \sum_{a \in \mathbb{A}} \| \boldsymbol{R}^* \boldsymbol{x}_a + \boldsymbol{t}^* - \boldsymbol{y}_a \|_2^2 \right]^{1/2},$$

where $(\boldsymbol{R}^*, \boldsymbol{t}^*)$, obtained via Kabsch alignment, is defined as

$$(\boldsymbol{R}^*, \boldsymbol{t}^*) = \underset{\boldsymbol{R}, \boldsymbol{t}}{\text{argmin}} \sum_{a \in \mathbb{A}} \| \boldsymbol{R} \boldsymbol{x}_a + \boldsymbol{t} - \boldsymbol{y}_a \|_2^2.$$

The alignment is recomputed at each forward pass and treated as constant during backpropagation. Here, $\mathcal{L}_g$ denotes the geometry loss, defined in the previous section.

Domain-motion steering. Proteins came from the DynDom (Qi et al.) dataset. We discarded sequences where both conformers came from a single PDB entry, entries that contain more than 1 protein conformation, conformer pairs being less than 90% identical, mismatched SEQRES and ATOM records, and sequences over 500 residues. We then further filtered on RMSD of the N/CA/C/O atoms being greater than $3\mathring{A}$, which matches the community metric of $\tau \geq 3\mathring{A}$ for a successful steering. This resulted in 340 usable proteins. Then, proteins were clustered with MMseqs2 (Steinegger and Söding) using cascaded easy-cluster at a 30% minimum sequence identity and 50% coverage. Easy-search was subsequently run to identify pairs that satisfied both criteria but were placed in different clusters, and affected clusters were merged. The training/validation/test split was then done based on clusters.

To define levels for this dataset, we defined level -1 to be the conformer more commonly sampled by BioEmu, and level +1 to be the conformer less commonly sampled by BioEmu. We make this distinction here because the conformers are not deposited in any particular order in the DynDom dataset, unlike with cryptic pockets and kinases which have explicit apo/holo and DFG-in/DFG-out labeling.

For domain-motion steering, the total objective is

$$\mathcal{L}_{\text{domain}} = \lambda_m \mathcal{L}_m + \lambda_g \mathcal{L}_g$$

where $\mathcal{L}_m$ and $\mathcal{L}_g$ are defined above. For $\mathcal{L}_m$, all residues are used compared to the pocket-context subset used by the cryptic pocket steering.

Kinase-state steering. Proteins came from the Kincore (Modi and Dunbrack) dataset. We discarded proteins that were not human, proteins generated from AlphaFold, proteins with no DFG motif label, and sequences with more than 500 residues. This left 61 usable proteins. We evaluated on our own 20 DFG-in and DFG-out pairs, as the set of proteins used in ConforNets (Lee et al., a) wasn't purely DFG-in and DFG-out pairs.

For kinase steering, the total objective is

$$\mathcal{L}_{\text{kinase}} = \lambda_m \mathcal{L}_m + \lambda_g \mathcal{L}_g$$

where $\mathcal{L}_m$ and $\mathcal{L}_g$ are defined above. For $\mathcal{L}_m$, only DFG-site and A-loop residues are used for the RMSD metric, as labeled by the Kincore dataset (Modi and Dunbrack).

A.8 Steering baselines

We used the same benchmark systems as curated in BioEmu (Lewis et al.), and as used in ConforMix (Richman et al.) and ConforNets (Lee et al., a), but with a mostly separate kinase set to ensure that all sequences have a deposited DFG-in and DFG-out state.

BioEmu baseline For BioEmu (Lewis et al.), we used the DPM sampler with 49 steps, queried the ColabFold server (Mirdita et al.) for MSA embeddings, and generated 300 structures per sequence for the success@100 metric. A random subset of these 300 structures was used for the success@5 metric.

ConforMix baseline We followed the paper and official codebase. We swept from 0.5 - 19.5 $\mathring{A}$, used the default twist strength of 15, and sampled 1 seed. This is the default of ConforMix (Richman et al.), but is different than what is reported by ConforNets' (Lee et al., a) comparison. We conduct the ConforMix benchmark with the Boltz version.

ConforNets baseline The numbers we report come directly from the ConforNets paper (Lee et al., a), using the distogram CDF MSE objective. Importantly, our benchmark has two fewer structures in cryptic pockets than theirs due to shipped benchmark information and residue coverage, and one more structure in domain motion again due to shipped benchmark information.

RMSD τ We follow ConforNets' RMSD τ benchmark for evaluation, where a structure is counted as successful if the aligned structure RMSD is below some threshold τ defined for each system. For

cryptic pockets $\tau = 1\mathring{A}$ (conducted on pocket residues specified by BioEmu (Lewis et al.)), domain motions $\tau = 3\mathring{A}$, and kinases $\tau = 2\mathring{A}$ (conducted on DFG-site and A-loop residues from Kincore (Modi and Dunbrack)).

B Additional results

B.1 Extended R_g steering figures

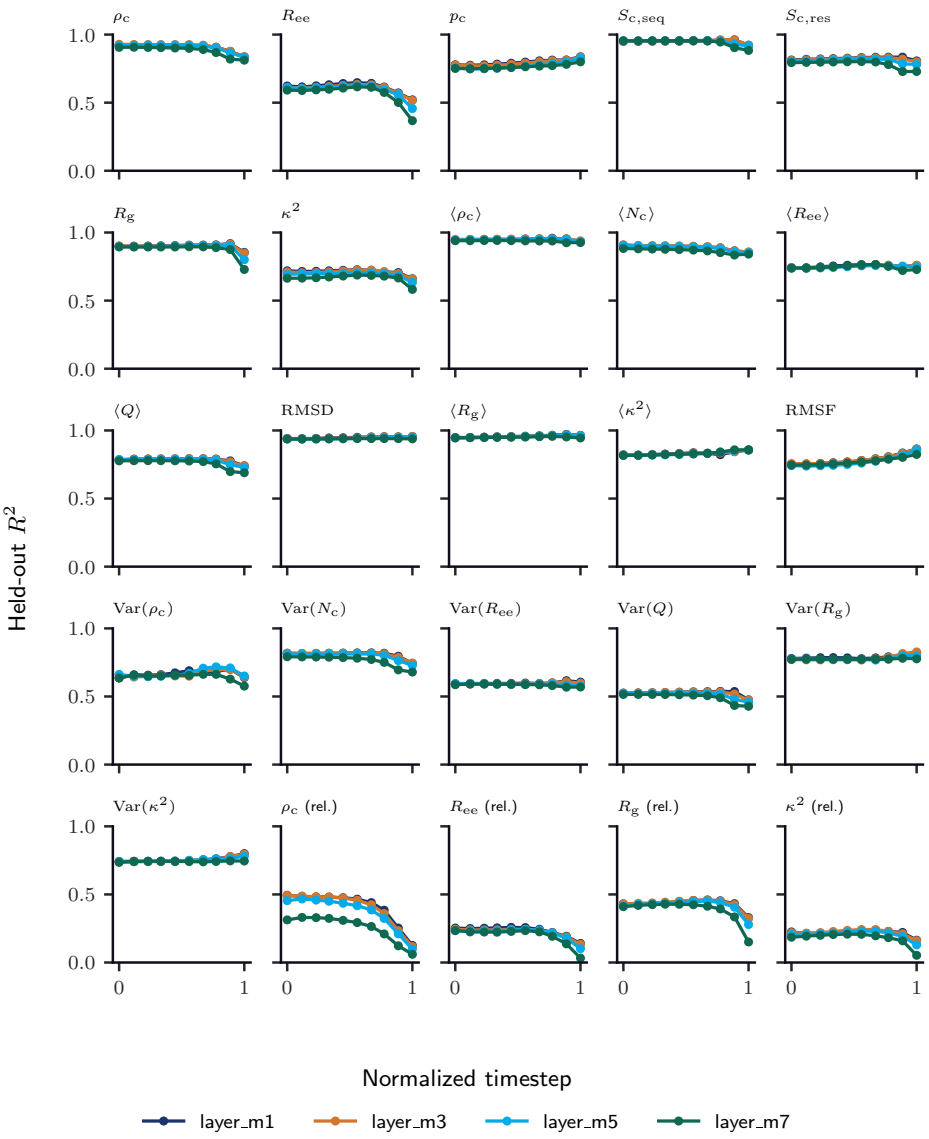

Figure A1: Raw 512 dimension activations ridge fit to biological properties and reported with R^2 using 5-fold cross validation across different denoising timesteps. Each line represents a different transformer layer the activations were taken from. Property abbreviations are as follows: contact density ρ_c, sequence contact entropy $S_{c,seq}$, residue contact entropy $S_{c,res}$, contact number N_c, relative shape anisotropy κ^2, end-to-end distance R_{ee}, mean contact probability p_c, local Q Q, a relative/normalized property (rel.), an ensemble mean of a property $\langle\rangle$, an ensemble variance of a property Var().

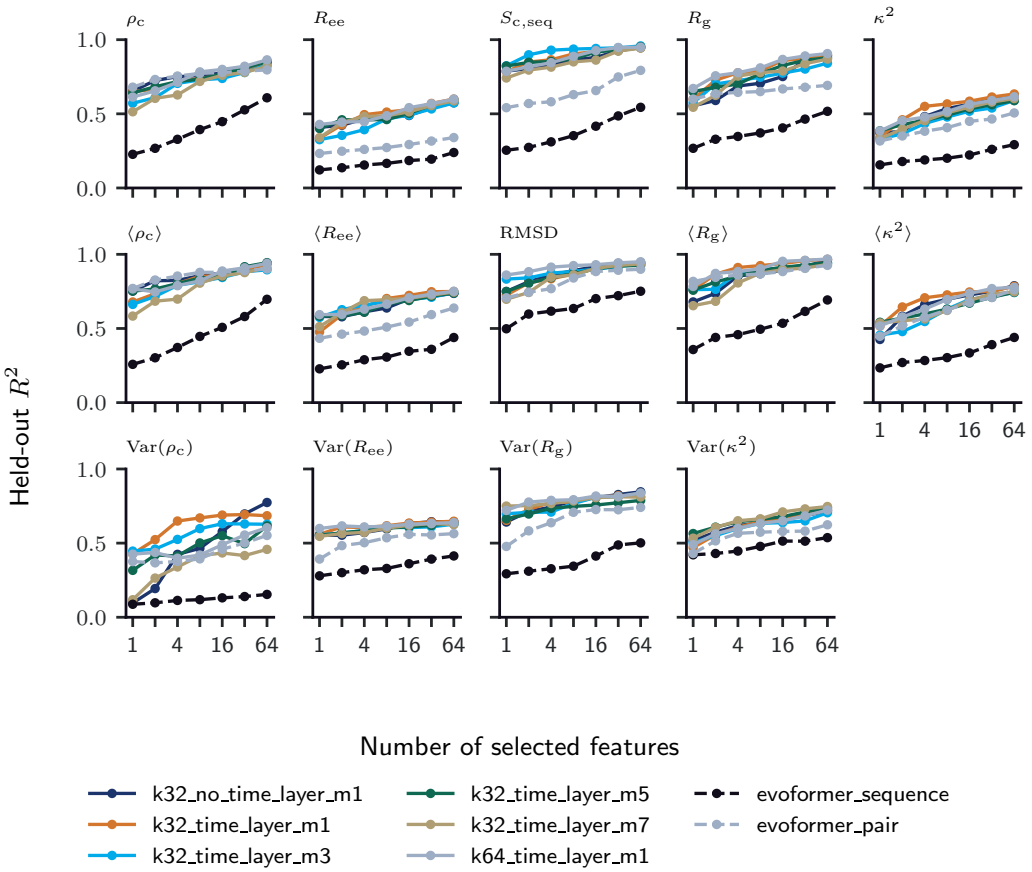

Figure A2: 5-fold cross validation R^2 values for varying numbers of SAE features ridge fit to biological properties, matched against Evoformer baselines. Property abbreviations are as follows: contact density ρ_c, sequence contact entropy $S_{c,seq}$, residue contact entropy $S_{c,res}$, contact number N_c, relative shape anisotropy κ^2, end-to-end distance R_{ee}, mean contact probability p_c, local Q Q, a relative/normalized property (rel.), an ensemble mean of a property $\langle\rangle$, an ensemble variance of a property Var().

The general trend seems to be that single-sample structural properties decrease in correlation as the timesteps become more noised, whereas ensemble properties tend to increase in correlation as the timesteps become more noised.

The base R_g adapter performs marginally better than most of the ablations. Initialization has little effect; removing the contrastive term reduces performance; removing the geometry term has little effect; $t = 0.005$ destroys structures; and directly concatenating the level into the input destroys learning (Fig. A3).

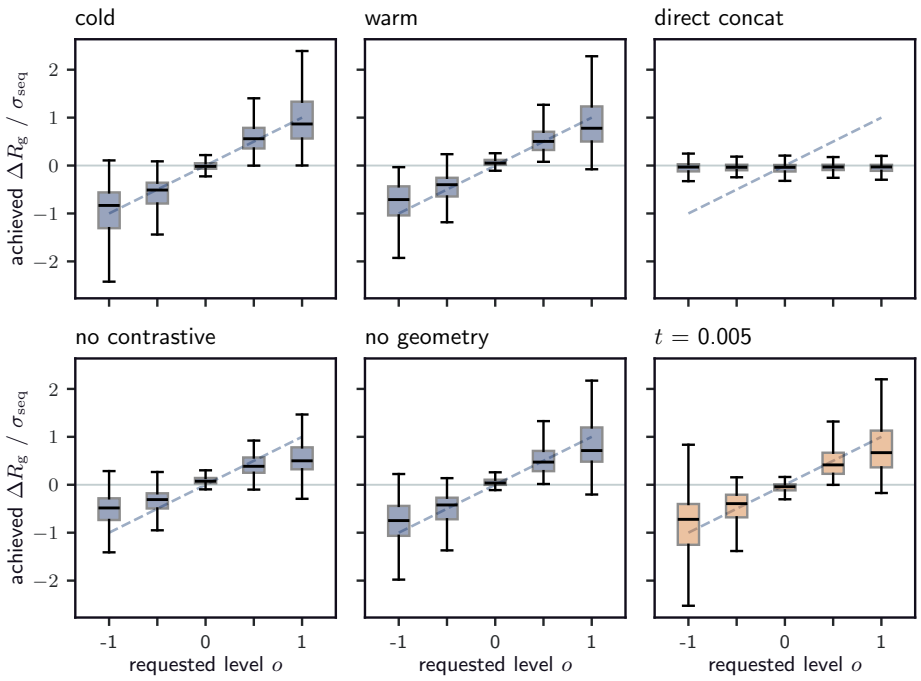

Figure A3: Box plot of achieved steering vs. requested steering across all of the steering ablations.

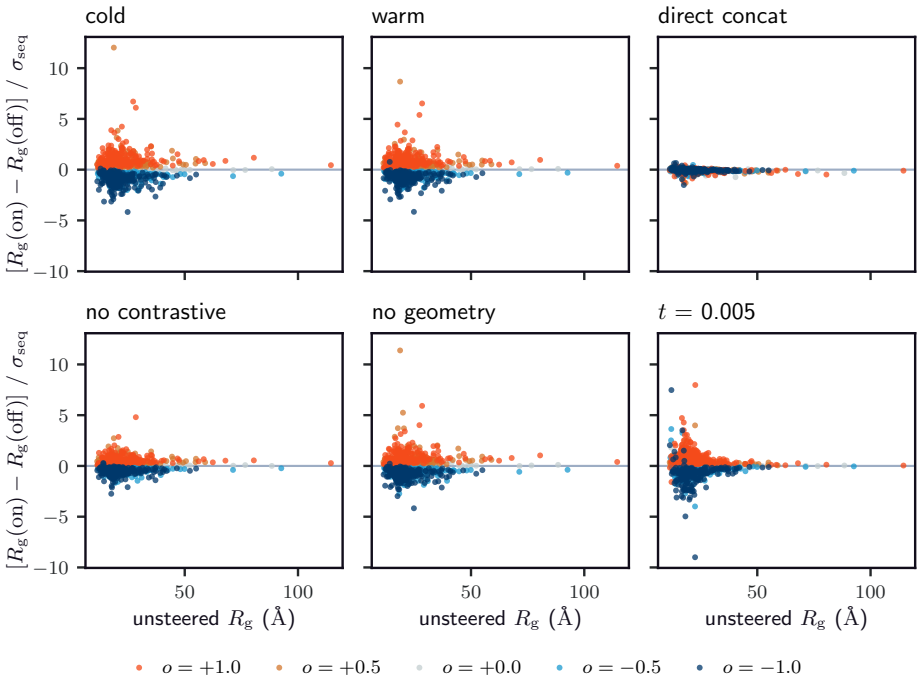

Figure A4: Steering response vs. R_g, colored by target level, across all of the steering ablations.

B.2 Extended cryptic pocket, domain motion, and kinase steering

Performance of each method at different B values for success@B. In some benchmarks our method performs better at lower B values, and in some cases it performs better at high B values (Table A1, A2). The choice of τ for each benchmark matters slightly, but not by too much (Fig. A5). In (Fig. A6, A7), we observe that although our method doesn't allow a large number of structures to pass the τ threshold if they couldn't before, almost all of the systems will move toward the targeted structure.

Table A1: Steering performance across conformational benchmarks for success@10, with standard error. ConforNets results are omitted here as they are not reported in the ConforNets paper.

Benchmark	N	BioEmu	ConforMix	Ours
Cryptic pockets (apo)	32	0.191 ± 0.063	**0.286** ± 0.077	0.214 ± 0.072
Cryptic pockets (holo)	32	0.331 ± 0.070	0.458 ± 0.078	**0.500** ± 0.082
Domain motions	44	0.591 ± 0.073	**0.620** ± 0.076	0.586 ± 0.068
Kinases (DFG-in)	20	0.766 ± 0.092	0.712 ± 0.092	**0.771** ± 0.090
Kinases (DFG-out)	20	0.479 ± 0.105	0.517 ± 0.107	**0.524** ± 0.104

Table A2: Steering performance across conformational benchmarks for success@50, with standard error. ConforNets results are omitted here as they are not reported in the ConforNets paper.

Benchmark	N	BioEmu	ConforMix	Ours
Cryptic pockets (apo)	32	0.252 ± 0.076	**0.312** ± 0.083	0.234 ± 0.074
Cryptic pockets (holo)	32	0.498 ± 0.080	0.560 ± 0.089	**0.590** ± 0.085
Domain motions	44	0.697 ± 0.072	**0.715** ± 0.084	0.697 ± 0.069
Kinases (DFG-in)	20	0.799 ± 0.092	0.812 ± 0.083	**0.824** ± 0.083
Kinases (DFG-out)	20	0.577 ± 0.106	0.593 ± 0.111	**0.625** ± 0.107

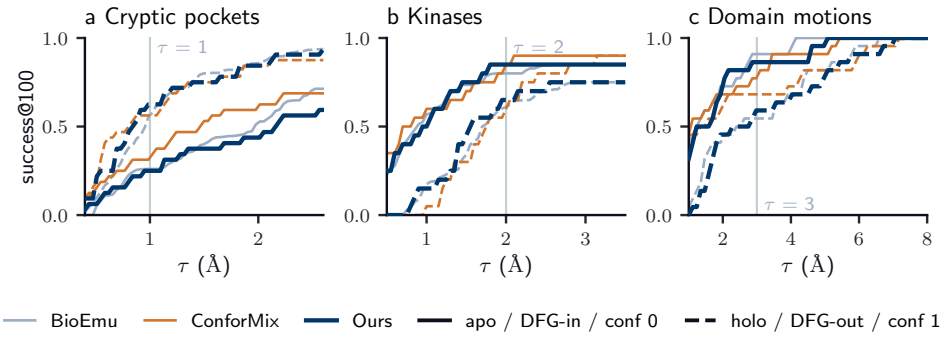

Figure A5: Success@100 as τ varies for different models.

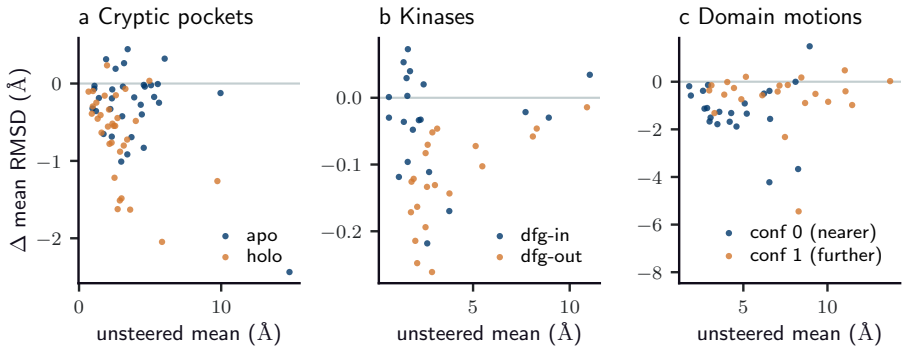

Figure A6: Scatter plot of benchmark proteins and their movement from steering.

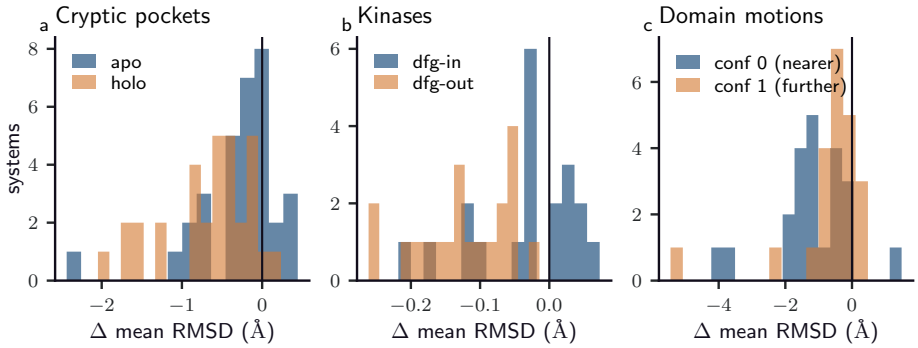

Figure A7: Histogram of benchmark proteins and their movement from steering.