DESTABILIZE TO STABILIZE: INFERENCE-TIME GUIDANCE FOR GENERATING MORE STABLE PROTEINS AND COMPLEXES

Anonymous authors
Paper under double-blind review

ABSTRACT

We aim to design more stable proteins by steering protein diffusion models such as BoltzGen. We hypothesize that generative models learn an implicit plausibility landscape around generated proteins and that this landscape can be used to study structural destabilization modes and corrective stabilizing sequence mutations. We introduce RipZipGuide, a training-free workflow based on inference-time guidance of diffusion models. Our destabilization technique operates on collective variables and identifies structural failure modes through progressive destabilization steering of generated proteins. A candidate protein or protein-complex is progressively destabilized over multiple noising-denoising iterations, with noise levels dynamically adapted to the candidate and with repulsive bias accumulated across iterations to discourage returning to previously sampled conformations, resembling metadynamics. Failure modes become observable as local unfolding events in nearby conformational basins. These regions are then patched/stabilized through targeted denoising using a bias optimizing for hydrogen-bonding. Case studies on a toy-model and realistic proteins demonstrate the universality, reproducibility, and utility domains of the method. We expect the primary benefit of the method to be in the speed-up of sampling proteins with custom properties without requiring any (re)training or fine-tuning of the base-model.

1 INTRODUCTION

Generative models have substantially expanded computational protein design, from unconditional backbone generation to motif scaffolding, binder design, and all-atom complexes (Watson et al., 2023; Ingraham et al., 2023; Stark et al., 2025). Once a plausible candidate is obtained, redesign may target properties such as affinity, specificity, solubility, expression, or catalytic activity while preserving its intended function. Stability is especially important for industrial proteins, which must withstand expression, purification, formulation, manufacturing, storage, transport, and application-specific operating conditions. Identifying and improving stability-limiting features is therefore a key objective in protein redesign.

Many protein foundation models, however, do not explicitly support conditioning for industrial requirements such as manufacturability or stability, as this requires accounting for features in training time. Such requirements can be addressed by steering a pretrained protein diffusion model at inference time (Nair et al., 2023). For proteins, physics-based objectives (Lai et al., 2026) and structural constraints (Utkarsh et al., 2025) can be used instead of classifier models.

Stability can be attributed to both local and global factors. Here, we focus on structural failure modes that appear during progressive unfolding and manifest as "local unfoldings" that precede global unfolding, but already may affect protein function. We propose an approach to efficiently generate proteins more prone to such unfoldings while preserving all conditioning supported by a base model. We focus on hydrogen-bonding failures, but other factors may include poor protein-core packing and lacking S-S bonds.

A generative protein model samples from a distribution of structures $p_{base}(X_0, c)$, where X_0 is an array with 3D positions of all protein atoms, c is the conditioning information. This distribution defines a pseudo-free-energy landscape $f_{base}(X_0, c) = -\ln(p_{base}(X_0, c))$. Generative models are

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.

known to be "reasonable" in a sense that minima of this landscape are often close enough to experimental structures of corresponding proteins to provide useful insights (Richman et al., 2025). We hypothesize that in addition to realistic minima positions in X_0 space, $f_{base}(X_0, c)$ reasonably approximates the true landscape near these high-probability basins, which corresponds to partially unfolded proteins. We propose to combine exploratory approaches with recent generative models that allow co-designing the structure and sequence (Stark et al., 2025; Geffner et al., 2025). Co-design capabilities of the base-model allow us to close the discovery-design loop to not only find conformational changes but also guide the sequence-generating regime to design mutation targeted at altering the identified conformational changes. Specifically, we focus on increasing the stability of generated proteins by identifying "weak-link" regions and then "patching" them with mutations.

We introduce RipZipGuide and use BoltzGen as the base-model. Our main contributions are:

1. **RipGuide** workflow: gradually rip proteins apart to find structurally-weak regions. The identified weak regions are often in less structured areas, as expected from physics, but the workflow provides more structural insight than simple metrics as pLDDT because only a fraction of such less-structured regions end up labeled as weak. The workflow is applied to standalone protein and parts of protein- or drug- complexes. The method is inference time, so does not require any (re)training. Any differentiable generative diffusion models should work as a base model (Abramson et al., 2024; Watson et al., 2023; Passaro et al., 2025).
2. **ZipGuide**: a complementary workflow to reinforce/patch weak regions by guided mutations. The mutations applied to a drug-complex preserved the drug binding potency, proving the base-model conditioning capabilities are preserved. The method is also inference-time, so no (re)training. The base model needs the capability to co-design structure and sequence (Geffner et al., 2025; Stark et al., 2025).
3. Compared to the base-model, an estimated increase of ~ 100 in efficiency of sampling structures with improved stability without significant violations of the generation conditioning of the base-model.

2 Background and Related Work

Protein redesign can benefit from overlapping the ability of inference-time guidance or pretrained generative models without retraining with conformational exploration methods applied to the rich structural priors encoded by protein generative models.

Protein generative and structure-prediction models. Diffusion and flow models generate plausible protein structures and sequences, including RFdiffusion, FrameDiff (Yim et al., 2023), Chroma, EigenFold (Jing et al., 2023), Genie 2 (Lin et al., 2025), Proteina (Geffner et al., 2025), and DiffDock (Corso et al., 2023), providing diverse structural priors for steering. Structure prediction has similarly progressed from AlphaFold2 (Jumper et al., 2021) to diffusion-based AlphaFold3 and Boltz-1 (Abramson et al., 2024; Wohlwend et al., 2024), alongside HalluDesign (Fang et al., 2025) and BoltzGen (Stark et al., 2025) which unifies design and prediction through specification of binding sites, covalent bonds, and structural motifs. CryoBoltz further steers Boltz-1 using cryo-EM density (Raghu et al., 2025), while recent work connects diffusion models with thermodynamic protein engineering (Li et al., 2025). While classifier and classifier-free guidance often require task-specific training (Dhariwal & Nichol, 2021; Ho & Salimans, 2021), training can be avoided if a "classifier" is not a model that requires training but a simpler expression, such as a physics-motivated fitness score. Many flavors exist (Bansal et al., 2024; He et al., 2024; Mammadov et al., 2026; Uehara et al., 2025; Utkarsh et al., 2025; Moitra et al., 2026). In proteins, steering has incorporated energy potentials (Gruver et al., 2023), reward guidance (Uehara et al., 2025), metadynamics-inspired biasing (Omidi et al., 2026), and embedding-space optimization (Li et al., 2026).

Physical and learned conformational exploration. Proteins occupy dynamic conformational ensembles, and conventional molecular dynamics (MD) (Fermi et al., 1955) can reach timescales beyond unbiased dynamics via enhanced-sampling methods. Common techniques include umbrella sampling (Torrie & Valleau, 1977), metadynamics and its variants (Laio & Parrinello, 2002; Invernizzi & Parrinello, 2020; Barducci et al., 2008; Pfaendtner & Bonomi, 2015), replica exchange (Sugita & Okamoto, 1999), accelerated MD (Hamelberg et al., 2004; Miao et al., 2015),

and weighted-ensemble or milestoning approaches (Zuckerman & Chong, 2017; Elber, 2020). Generative models instead learn conformational distributions directly, from Boltzmann Generators (Noé et al., 2019) and disordered-protein models (Janson et al., 2023; Janson & Feig, 2024) to diffusion and flow-based ensemble models (Lu et al., 2024; Fan et al., 2024; Jing et al., 2024a; Jin et al., 2025; Zheng et al., 2024; Lewis et al., 2025; Liu et al., 2025; Jing et al., 2024b). More recently, enhanced-sampling ideas have entered diffusion inference due to many-samplers techniques allowing quantitative sampling from skewed distributions: Twisted Diffusion Sampling (TDS) (Wu et al., 2023) uses Sequential Monte Carlo (SMC) (Doucet et al., 2001) with many "samplers", and Feynman-Kac (Singhal et al., 2025) sampling is another many-samplers approach. ConforMix employs TDS to combine biasing, resampling, and reweighting (Richman et al., 2025) to explore conformational space and obtain free-energy landscapes. Related methods introduce collective-variables (CVs)-, repulsion-, or restraint-based guidance (Nam et al., 2026; Ohnuki & Okazaki, 2026; Lam et al., 2026; Hori et al., 2026).

A concept of a collective-variables (CV) is important for both exploration and diagnosis of proteins. Often, a CV is a scalar function of the system state X. A well-chosen CV can greatly help in diagnosing the system because applying external forces along this collective degree-of-freedom (DOF) can quickly lead to rare barrier-crossing events along this DOF. Efficient exploration without prior knowledge is often complicated. Our approach is to devise general-purpose CVs that are expected to promote exploration for a wide class of proteins regardless of details such as specific protein sequences.

3 METHODS

3.1 DIFFUSION MODELS AND DPS-LIKE "ENERGY" GUIDANCE

The base-model BoltzGen utilizes the EDM sampler (Karras et al., 2022) that models the flow

$$\mathrm{d}X_t = -\,\boldsymbol{\nabla}_x \ln[p(x,\sigma(t))]\Big|_{x=X_t}\dot{\sigma}(t)\sigma(t)\mathrm{d}t = -d(X_t,\sigma(t))\mathrm{d}t \tag{1}$$

with $\sigma(t)=t$, drive-vector $d(x,\sigma)$ and the score estimator $\boldsymbol{\nabla}_X \ln[p(X,\sigma(t))] \approx (D_\theta(X,\sigma(t)) - X)/\sigma^2(t)$ where $D_\theta(X,\sigma) \approx \mathbb{E}[x_0|x_{t(\sigma)}=X,\sigma]$ is the trained denoiser. For us, X_t is the system-state of $N_{atoms}\times 3$ coordinates at "time" t. Using the sampler with the trained denoiser D_θ yields the base-model distribution $p_{base}(X_0)$. We then aim to sample a modified biased distribution without retraining the base-model D_θ:

$$p_{guided}(X_0,c) = p_{base}(X_0,c)[\exp(-u_b(X_0))/Z_{bias}(c)], \tag{2}$$

where c is the conditioning supported by the base model, $u_b(x)$ is the "biasing potential" and $Z_{bias}(c) = \int p_{base}(X_0,c)\exp(-u_b(x))\mathrm{d}x$. For that, we employ the "Energy-guidance" (Zhao et al., 2022; Yu et al., 2023) approach that uses $\sigma(t)=t$ and modifies the drive vector-field as

$$d_{guided}(t,X,c) = d_{base}(t,X,c) - \sigma(t)\boldsymbol{\nabla}_X \ln \mathbb{E}\Big[e^{-u_b(X_0)}|t,X,c\Big] \tag{3}$$

The exponential average is hard to compute. The SMC (Doucet et al., 2001) with $K \gg 1$ samplers in TDS (Wu et al., 2023) is an asymptotically correct way to estimate it for enhanced-sampling (Richman et al., 2025). However, we can use a cheaper sampling similar to DPS (Chung et al., 2023)

$$d_{corr}(t,x,c) = d_{guided}(t,X,c) - d_{base}(t,X,c) \approx \sigma(t)\boldsymbol{\nabla}_X[u_b(D_\theta(t,X,c))] \tag{4}$$

because our workflow does not require quantitative free-energy estimates enabled by TDS. This is just a 1st-order cumulant expansion of equation 3. Auto-differentiation allows to avoid materializing a $3N_{atoms}\times 3N_{atoms}$ Jacobian of the denoiser. Also, see Section A for analysis of guidance terms beyond 1st-order cumulant expansion. We use equation 4 to apply both destabilizing (Figure 1.2) and patching (Figure 1.4) biases described below.

3.2 RIPGUIDE: DESTABILIZATION GUIDANCE TO DISCOVER UNFOLDING PATHWAYS

The main goal of this step is to discover "accurate" unfolding pathways of a given structure X_{init} in a given base-model free-energy landscape $f_{base}(X_0,c)$. We aim to produce a series of progressively

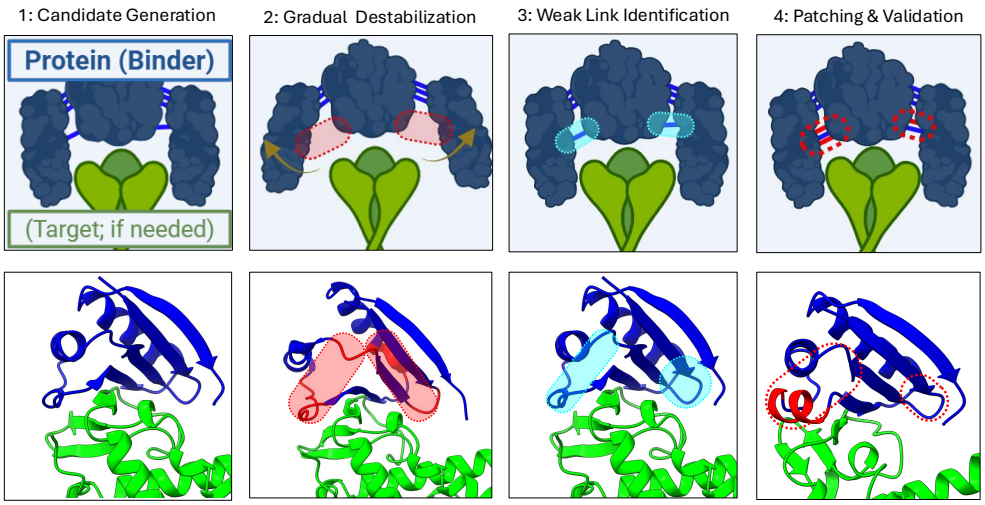

Figure 1: RipZipGuide identifies and reinforces weak links in proteins and complexes. (1) A protein to be stabilized (blue) and its binding target (green). The workflow preserves the base-model conditioning such as binding sites. (2) With the AA sequence fixed, RipGuide progressively disturbs the original structure X_{orig} by moderately noising it and denoising with the accumulating bias $u_b = u_{destabilize}$ via equation 4. Uniformly increasing stress on native contacts of atoms breaks the weakest links first (red). (3) DBSCAN clusters frequently and strongly disturbed atoms into weak-link regions (cyan). (4) ZipGuide re-noises X_{orig} and denoises with a patching bias $u_b = u_{patch}$ to introduce mutations (red) that reinforce hydrogen bonding in these regions.

Algorithm 1: RipGuide high-level overview. Complete version: Algorithm 2 in Section C.1.

$n \leftarrow 0$, $X_0 \leftarrow X_{orig}$, $u_0^{bias}(x) \leftarrow 0$, $p \sim 5$, $e_{noise,1}$ ~0.01A;
$NC \leftarrow$NativeContacts[X_{orig}, $R_{cut} \sim$ 4A, Amino-acid sequence distance > 2];
repeat
  $n \leftarrow n + 1$;
  $X_n^{noise} \leftarrow X_{orig} + e_{noise,n}\mathcal{N}(0,1)$; `// Step 1: noise original structure`
  $u_n^{bias}(x) \leftarrow u_{n-1}^{bias}(x) + u_{repel}^{bias}(x, X_{n-1}, NC)$; `// accumulate repulsive bias`
  $X_n \leftarrow$ denoise X_n^{noise} via the base-model steered with $u_n^{bias}(x)$; `// See Step 2`
  $e_{noise,n+1} \leftarrow$ mean$^{1/p}$[$(RMSD_p^*)^p$ history of last ~ 10 samples]; `// See Step 3`
until [A fraction of ~ 10 last X_n-s with $RMSD_p^*(X_n, X_{orig}) > RMSD_{max} \sim 6A$ exceeds a $\nu_{max} \sim 0.5$ threshold, so unfolded states are sampled often, implying folded basins filled up];
$w \leftarrow$ per-atom native-contact loss (PN-scores) averaged over collected structures $\{X_n\}$;
weak-link regions $\leftarrow$ DBSCAN($X_{orig}, R_{cut} \sim 4A$, weights $= w$, include top $\sim 10\%$ atoms);

disturbed structures starting from X_{init} that discover basins nearby X_{init} in the order sufficiently similar to the order that these basins would be visited/discovered by the corresponding real protein during its unfolding. We choose a mechanism arguably closer to thermal destabilization: near-independently apply a growing "stress" to all native contacts in the given protein until the weaker contacts break. Weak-links discovered in this way are expected to be generally less resistant to a wide range of disturbances, so the weak-links we identify here are expected to be more broadly useful than just for thermal destabilization. Algorithm 1 summarizes RipGuide, Algorithm 2 in Section C.1 gives full details.

1. At a step n, noise X_{orig} to a $e_{noise,n}$ level as $X_n^{noise} = X_{orig} + e_{noise,n} \cdot \mathcal{N}(0,1)$. Noise-level $e_{noise,n}$ is dynamically chosen to progressively explore the phase-space because, assuming

X_{orig} is near a local optimum of the base-model distribution, only states within $\sim e_{noise,n}$ from X_{orig} are easily accessible when denoising from X_n^{noise}.

2. Denoise the "blurred" structure X_n^{noise} into a new structure X_n using the base-model + guidance repelling from the states already sampled before via $u_n(X) = a_{metaD} \sum_{k=0}^{n} u_1(X, X_k, NC)$. Here $NC = \text{NativeContactAtomPairs}(X_{orig})$ (see Section C) and a_{metaD} is a dimensionless bias strength. This is the meta-dynamics-inspired step. The CV should represent a generic destabilization, so an RMSD-like CV was chosen as a proxy for that: $u_1(X, X_n, NC) = \sum_{(i,j)\in NC} \exp\left[-(|r_{ij} - r_{n,ij}|/\sigma_d)^2\right]$, where σ_d is a characteristic bias range.
3. Progressively disturbed structures X_n from many such noise-denoise steps are accumulated and the total bias for the next denoising is $u_n(X) \sim \sum_{k=0}^{n} u_1(X, X_k, \dots)$. Noise-level $e_{noise,n}$ is dynamically chosen as a mean over the $RMSD_p^*$-s of N_{last} latest-sampled states from X_{orig}, where $RMSD_p^*(X_1, X_2, \text{NativeContacts}) = \text{mean}^{1/p}_{\text{NativeContacts}}(|r_{1,ij} - r_{2,ij}|^p)$. The mean and $RMSD_p^*$ are p-powered, i.e. $e_{noise,n} = \alpha_{rescale} \left(\text{mean}_{[-N_{last}:]} \left[\left(RMSD_p^* X_{orig}, X_k, \text{NativeContacts})\right)^p\right]\right)^{1/p}$ so that p can tune between emphasize of the mean ($p \sim 1$) or on the maximum deviation ($p \gg 1$). Values $\alpha_{rescale} > 1$ create an extra "drive" to explore and was found to help. The exact algorithm is in Section C.1.

The proposed loop resembles metadynamics, but with modifications. Regarding the CVs used, the first difference from the regular position-based RMSD is that we are using deviations of pair-distances (instead of atom-positions) because we want the disruptions to focus on breaking non-covalent contacts in the native structure instead of e.g. domain-reorientation as in (Richman et al., 2025). Second, we are applying per-DOF bias (i.e. many 1D biases) instead of exploring full high-dimensional phase-space of all pairwise distances. Such per-DOF biasing is known as parallel-bias meta-dynamics and was shown to be more efficient for exploration than "regular" enhanced-sampling that explores full phase-spaces of CVs (Fu & Pfaendtner, 2018). For us, parallel-biasing decouples "stresses" applied to different atom-pairs. Such decoupling is desirable for identifying weak-links since stresses accumulate uniformly among native contacts and in "time", thus the first bonds to break are expected to be the weakest. An important difference from metadynamics with MD sampling is that MD is forced (by high- frequency motions) to have a small timestep and thus is forced to base its next step on its previous state to reach any states meaningfully different from X_{orig}. Using diffusion sampling avoids small time-integration steps (while still using many "small" steps to denoise each sample), thus we can avoid changing the initial state but noise from X_{orig} every time to a sufficiently large $e_{noise,n}$.

The loop continues until we start registering a high frequency of structures with high $RMSD_p^*(X, X_{orig}, \text{NativeContacts}) > RMSD_{max} \sim 6$Å . Having an ensemble of progressively disturbed structures $\{X_n\}$ and assuming the width W_{well} of the "potential well" around X_{orig} is smaller then the inter-well distance to neighboring wells, we can choose states from neighboring wells (as opposed to variations inside the original well and also opposed to significantly blown-up systems) based on $RMSD_p^*$, where choosing appropriate p helps to make the relevant peaks in the total distribution of $RMSD_p^*$ more separable (see Section B.3).

Having the states that are (mostly) from potential basins near-by from the X_{orig} well, we can weigh each atom in X_{orig} proportional to its contribution into $RMSD_p^*$-s of these states from X_{orig} as a per-atom measure of how many native contacts each atom loses (Section C.1), which refines "patching-necessity score" (PN-score). Then, clusters of atoms with high PN-scores would be the "weak-link groups" that need patching.

3.3 ZIPGUIDE: PATCHING GUIDANCE

Each localized weak-link group and residues with any atom therein are considered for patching/redesigning. X_{orig} is repeatedly noised and re-generated, but now with a different bias that aims to increase structural stability around the identified weak-link. It follows a 2-step process:

1. At each step n, the structure is re-noised as $X_n^{reseq} = X_{orig} + e_{reseq}\mathcal{N}(0, 1)$. Noise magnitude on the order of a typical amino-acid size $e_{reseq} \sim R_{AA}$ was found to work best. It makes sense because the base-model BoltzGen infers amino-acid (AA) types from superposing non-backbone

atoms on backbone atoms in a predefined ways different for different AAs, thus the noising should be bigger than a typical AA ($\geq R_{AA}$) to "blur" AA types but not much bigger than the protein ($\lesssim R_{protein}$) to preserve the coarse structure of X_{orig}.

2. The X_n^{reseq} is denoised with the base-model + guidance to "stabilize" weak-link regions. We chose a simple proxy of H-bonds inside the weak-link region or between weak-link and non-weak-link atoms. The base-model does not have explicit hydrogens, so we infer H-bonds from heavy-atom positions as following:

$$\mathcal{U}_{HB}(X) = -u_{HB} \sum_{(i,j)\in \text{HB_BB_pairs}} \exp\left[-(d_{N_i,O_j} - d_{eq})^2/2\sigma_{dist}^2 - (1+\cos(\theta_{C_j-O_j\cdots N_i}))/\sigma_{angle}^2\right] \quad (5)$$

where $(i, j) \in$ HB_BB_pairs pairs an amide N with a carbonyl O, exactly one of which belongs to a designable residue. We are only addressing backbone H-bonding because the base-model only guarantees fixed atom-types for backbone atoms and atom-types are necessary for H-bonding.

4 Experiments and Results

4.1 RipGuide for a Synthetic 2D toy example of destabilization

It is hard to systematically explore unfolding pathways of real proteins, so we first validate our destabilization approach on a synthetic generic 2D landscape. The main goal of this toy-model is to confirm that the proposed destabilization mechanism discovers "accurate" unfolding pathways: physically-realistic basins near the initial state X_{init} are discovered in physical order (Figure 2c). The landscape consists of a wide global well and several smaller wells near its minima (Figure 2a).

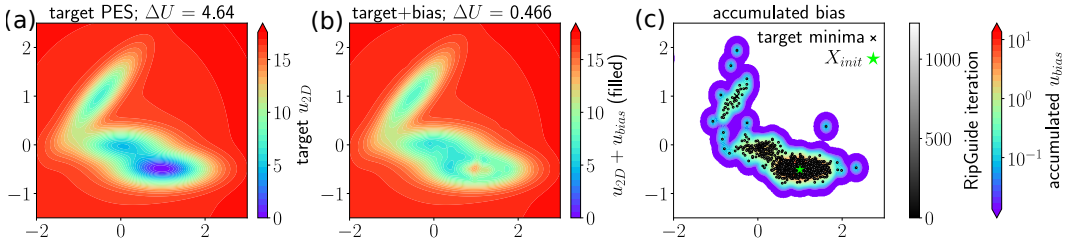

Figure 2: (a) The original "potential landscape". ΔU is the difference in energies between minima at (0;0) and (1;-0.5) (b) Filled-up landscape at the time of "blow-up" == end sampling (c) rainbow: the bias potential; black → white dots: samples points. Original basin: has black points → next basin has gray ones → next ones has white. So, the discovery of basins is in the right order.

The global well represents general loss of structure in a typical unfolding event, while smaller wells represent the "ground state" basin and a few less stable but still partially folded conformation basins. We use a Muller-potential-inspired landscape (Section B.1). A small diffusion model was trained on this landscape. We then initiate guided exploration in the deepest basin and investigate how well the nearby basins are discovered depending on the bias parameters, noising dynamical schedule and more. The resulting exploration "trajectory" (Figure 2c) and the biased free-energy landscape (Figure 2b) are shown at the point sampling was stopped. Details of experiments with this toy model are provided in Section B. From these experiments, we draw several insights. The referenced appendix figures are not needed for understanding the insights, only to give the data evidence supporting them.

1. First-order guidance follows predicted asymptotic noise scaling (Figure 8).
2. Adaptive noise without accumulated bias does not produce nearest-first discovery (Figure 5).
3. Large initial noise discovers basins but loses nearest-first discovery ordering (Figure 12).
4. Per-DOF biasing preserves the predicted scalings (Figure 9) and has an optimal regime where it results in only tolerable distortions to basins' geometry (Figure 10).
5. Repulsion must be deposited through small increments because the blow-up time scales as $\sim 1/a_{metaD}$, so strong bumps (high a_{metaD}) produce rapid blow-ups (Figure 6).
6. An effective bias bumps' width is comparable to the baseline fluctuation scale, whereas an overly broad bias fails to exclude visited states selectively (Figure 11).

4.2 RIPGUIDE FOR REAL PROTEINS

We found the following parameters to work the best: $\sigma_d = 1.5A$, which makes sense as $\gtrsim \sqrt{B_{loose}/8\pi^2} \sim 1.1A$ for a $B_2 \sim 100A^2$ typical for volatile atoms. Bias strength $a_{metaD} = 5 - 10$ was a good balance between destabilization speed and results reliability.

The regions identified as more volatile are less structured, as physically expected: Representative "destabilization trajectories" for a standalone protein and a protein complex are shown in Figure 3. Both unperturbed structures (Figure 3a,d) have more structured as well as less structured regions. Thicker red regions highlight regions with more native-contact disruptions (PN-scores) on average over the whole "trajectory". Early-stage snapshots (left on Figure 3(b,e)) are of near-original structures and they show high-PN-score parts are deeply embedded into less structured regions. This gives structural insight additional to simple features like pLDDT (Section C.2).

Destabilization "trajectories" resemble physically-motivated unfolding pathways: RipGuide does not have physical time and each new sample is obtained from the noised original structure. Still, frames on both Figure 3(b,e) visually resemble protein unfolding trajectories. We think this is because RipGuide similar to quasi-equilibrium evolution of a gradually heated structure since both physical heating and accumulation of repulsive bias make the native free-energy basins effectively shallower.

Destabilization of a binder in a complex preserves the base-model conditioning on binding the target: The "trajectory" in Figure 3e shows a gradually disrupted binder, but the model still is able to find binder conformations that most-plausibly bind the target because the conditioning on target binding is active throughout the whole destabilization process. To accommodate such binding, the target also sometimes undergoes moderate structural changes. The coarse target structure is preserved during destabilization, even with severely broken binder. Binding preservation for drugs is addressed in Figure 4.

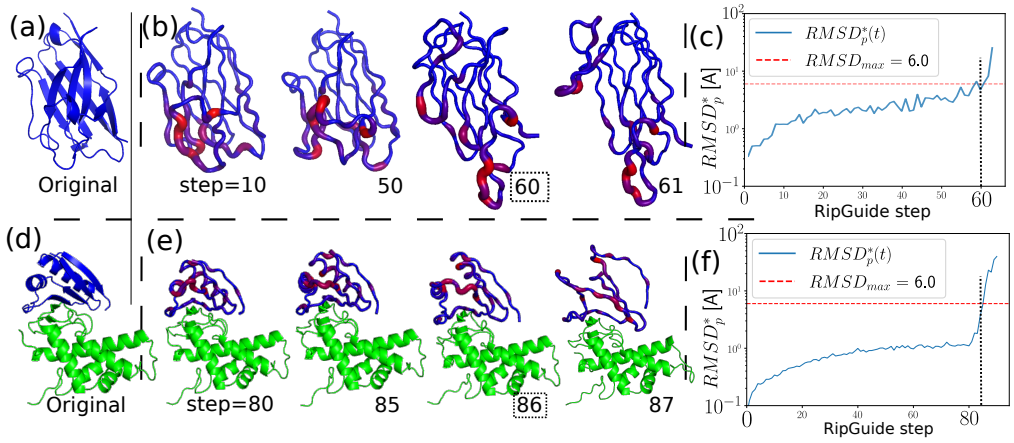

Figure 3: **(top) Stand-alone protein being disrupted:** (a) An unperturbed "original" structure of the protein to be disrupted (b) Representative progressively disrupted structures. Red thickness corresponds to the regions that end up being more mobile on average over the whole trajectory. Dashed box step corresponds to the vertical dashed line in (c). (c) "Evolution" of the Native-contact distance-$RMSD^*_{p=5}$ during the disruption bias accumulation. Red dashed line is a threshold so that overshooting it too often is the criteria for convergence. **(bottom) A mzb1-binder in a complex is being disrupted:** (d) The unperturbed structure (binder is blue, target is green). (e) Representative structures of the progressively disrupted binder still conditioned on binding the target. The binding is generally preserved, at least before severe disruptions. (f) Native-contacts distance-$RMSD^*_{p=5}$ of the binder protein.

Per-atom PN-scores and their clusters correspond to protein physical structure: Stochasticity is involved in RipGuide, but the "physical" PN-score that each atom receives is statistically reproducible as evident from how similar blue→red colorings are on Figure 4(a)-vs-(b) and (e)-vs-(f). Given per-atom PN-scores, the weak-link regions are obtained by clustering atoms via DBSCAN

with weights=PN-scores, $R_{cut} = 4$ Åand N_{min} chosen to have approximately 10% of atoms in the weak-link regions. Resulting clusters, which are the weak-link regions to be patched, are also reproducible, as seen from the similarity of dark-cyan stick-regions on Figure 4(a)-vs-(b) and (e)-vs-(f). See Section C.2 for statistics and explanations of patterns in PN-scores. Finally, the location of identified weak-link regions makes biophysical sense as less-structured regions are chosen.

4.3 ZIPGUIDE FOR REAL PROTEINS

Given the weak-link regions (Dark-cyan in Figure 4(a,b,e,f)), we perform noise-denoise cycles with a patching bias Equation 5. We found noising to $\sim$ 19Å and denoising with $u_{HB} = 4.0$ to be optimal (Table 1). Stronger biases disrupt the base-model and weaker biases give smaller effect. The $\sigma_{dist} = 1.0$Å and $\sigma_{angle} = 30°$ were found to be optimal. Mutations of weak-link regions of patched structures are shown in Figure 4(c,g) in red (mutated AAs) and orange (AAs with H-bonds to mutated AAs). The count of H-bonds including the mutated AAs typically goes up (Table 1). This is a proxy of general-purpose stability that should hold against a relatively wide range of thermal or chemical disruptions. Many physics-based stability tests exist via molecular-dynamics methods (review (Gonzalez et al., 2022)), however, such tests would be hard to interpret because a structure near-optimal for the "force-field" learned by BoltzGen may not be optimal for a physics-based forcefield used in MD. Additionally, realistic denaturation temperatures often change no more than $5 - 10°$C from local mutations (Musil et al., 2025) and require special engineering involving charge interactions (not included in our simple bias) to go beyond 20°C in only special cases (Fu et al., 2010). In-silico measurements with accuracy $\sim 5°$C are difficult (Pudžiuvelytė et al., 2024) so we use our destabilization approach to avoid the forcefield-mismatch and ensure self-consistent validation. We run identical destabilization regimes of the original and the patched structures with 10 random seeds and measure how long it takes to degenerate. Both MZB1 and Rucaparib binders (Figure 4(d,h)) increase in the number of destabilization steps until blow-up.

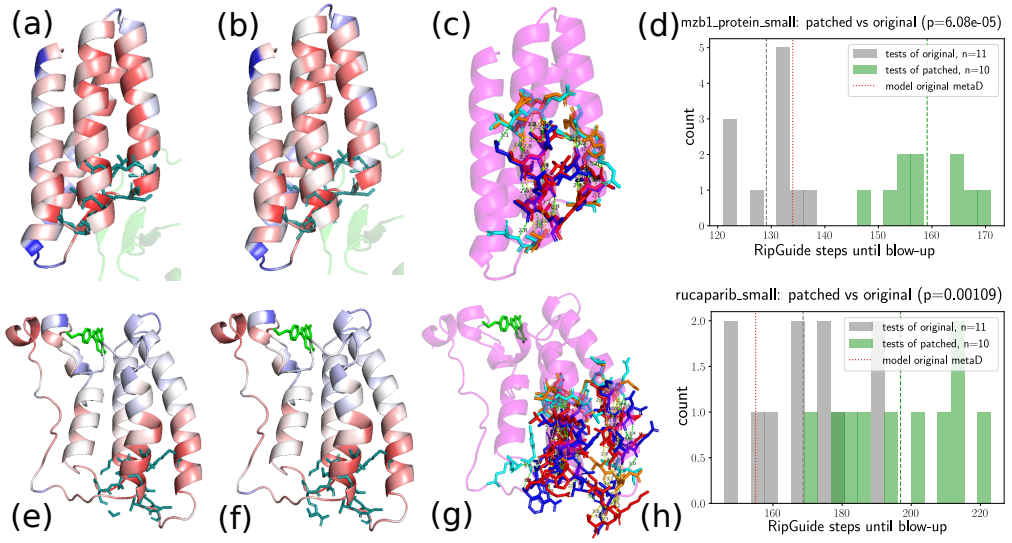

Figure 4: (a,b) Clustering of per-atom PN-scores identifies weaker regions. Colorings of (a) and (b) are close, proving the reproducibility of PN-scores. Dark-cyan sticks show clusters of high PN-scores as regions labeled for patching. (c) Original vs mutated regions. Magenta: full structure of the original binder. Red: AAs from weak-link regions that end up mutating. Orange: AAs that bond differently before and after mutations. Blue: AAs at the same positions as red ones, but in the original structure before mutations. Cyan: original AAs at the same positions as orange AAs. (d) Distribution of "durations" of destabilization "trajectories" for the original (gray) and patched (green) structures. p = P[data-mismatch|same-stability]. (e-h) Same as (a-d), but the target is Rucaparib drug (green). Binding potency is not significantly affected: from Botz2 estimations, $\Delta \log_{10}(IC_{50}) = 0.16 \pm 0.09$.

Table 1: Patching helps for low initial sampling. Relative change in the number of H-bonds $\delta_{X,ref} = N_{HB,X}/N_{HB,X,ref} - 1$, where X can be "all" (all H-bonds in the structure) or "unstab" (H-bonds including the weak-link region) and ref can be "base" (BoltzGen→BotlzIF→Boltz2) or "no-bias" ($u_{HB} = 0$ patching→BotlzIF→Boltz2). N_{init} is the number of initial candidates sampled from $base$. All relative differences are in %.

System	N_{init}	u_{HB}	$\frac{t_{patched}}{t_{orig}} - 1$	$\delta_{all,base}$	$\delta_{unstab,base}$	$\delta_{all,no-bias}$	$\delta_{unstab,no-bias}$
Rucaparib binder	10	4.0	16.7 ± 4.5	2.5 ± 2.2	30.7 ± 1.5	7.1 ± 1.0	19.7 ± 1.0
MZB1 binder	10	4.0	17.9 ± 2.4	54.4 ± 9.0	61.2 ± 2.5	8.0 ± 1.3	40.6 ± 2.2
MZB1 binder	1000	4.0	-3.1 ± 2.7	-7.2 ± 2.6	-0.9 ± 0.7	1.7 ± 0.6	2.3 ± 0.6
Penguinpox NB	1000	0.7	-57.7 ± 1.6	-14.1 ± 2.3	-5.7 ± 1.3	3.5 ± 0.8	13.0 ± 1.4
PDL1 Fab	1000	0.7	-4 ± 12	-6.5 ± 3.4	-8.3 ± 1.1	1.8 ± 0.9	2.4 ± 1.1

Table 1 shows that sufficiently strong patching guidance results in significant gains in both H-bond counts (especially near weak-link regions, see $\delta_{unstab,base}$ column) and resilience to destabilization ($t_{patched}/t_{orig}-1$). Last 2 columns are "controls" that H-bonding increased due to patching bias and not just resampling. We note that $\delta_{unstab,no-bias} < \delta_{unstab,base}$, which means some improvement can be gained from resampling even without bias. This is likely because the original structure was chosen from only 10 candidates, so it may be not close enough to a local optimum, so resampling around that structure mostly samples closer to the optimum.

Both successfully patched examples in Figure 4 had only 10 initial BoltzGen candidates generated instead of the recommended $\sim 10^{4-5}$. The structure for the RipZipGuide pipeline was selected as top1 according to default base-BoltzGen filtering. Increasing the number of initial candidates to 1000 and attempting RipZipGuide on the top 1000 was less reliable in producing any stability improvements. During patching, 30 patched structures were generated, which approximately matched wall-time required to sample 10 initial candidates since initial denoisings are done from much larger noise. The corresponding RipGuide also takes a similar wall-time. Improving patched structures also proved less reliable, perhaps due to saturation of local stabilizing modifications. Thus, we believe that structures sampled by ZipGuide-BoltzGen are within distribution for base-BoltzGen, just skewed towards more H-bonded structures. This can be an advantage since base-BoltzGen had extensive wet-lab-grade verifications. So, we were able to sample H-bonding-saturated structures within wall-time of $\sim$ 10 initial BoltzGen samples instead of $\sim$ 1000, which implies a $\sim$ 100 speedup if our goal is specifically H-bonding optimality.

The work can be expanded in several directions. Using a more complete model of protein stability for patching is fruitful for controlled mutations. More thorough investigation of structural degradation induced by the patching in some cases would clarify the applicability domain of the method. Untangling the split of stabilization effects from no-bias resampling (which purely explores local basins via regenerations cheaper than full denoising) vs patching bias. Perhaps no-bias resampling can be applied for proteins that are too delicate for strong patching.

5 Conclusion

We introduced RipZipGuide, a training-free, inference-time workflow that steers a pre-trained all-atom protein diffusion model, BoltzGen, towards stability, in particular, reinforced H-bonding network, which was not encoded as a conditioning variable during training. RipGuide progressively destabilizes a candidate protein with a metadynamics-inspired, per-degree-of-freedom bias and adaptive noise, exposing reproducible weak-links that are then clustered to identify failure regions. ZipGuide then noises and denoises with a hydrogen-bond-optimizing bias around these regions, producing mutated candidates with improved stability. Base-model conditioning capabilities are largely preserved for protein-protein and protein-drug complexes. For the limited set of studies that we conducted on real world proteins, we observed a speed-up of approximately 100x compared to repeated sampling of the base distribution and subsequent filtering. While the focus of our work was to look for hydrogen bonding, we believe this approach can be adopted for other industrial problems also.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks:

- Improving grammar, readability and conciseness: In the background and introduction section we used AI tools to integrate input from co-authors and to reduce the total content to fit the required page budget. We did, however, carefully verify the generated text to ensure it is consistent with the original content generated manually.
- Retrieval and Discovery: We started from a set of papers that we were aware of as a part of the background. This was augmented using AI tools to scan for related papers that we were not originally aware of. The additional references were however manually verified for applicability and discarded if spurious or if they didn't fit the scope.
- Research ideation and execution: We used AI tools as a coding support for the experimentation. However, we take full ownership of the algorithm and for devising the experiments.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, Jun 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, et al. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/dfbb3d1807b21dadee735eb75069ada4-Paper-Conference.pdf.

Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Phys. Rev. Lett.*, 100:020603, Jan 2008. doi: 10.1103/PhysRevLett.100.020603. URL https://link.aps.org/doi/10.1103/PhysRevLett.100.020603.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi S. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kKF8_K-mBbS.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Arnaud Doucet, Nando de Freitas, and Neil J. Gordon (eds.). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001. ISBN 978-1-4419-2887-0. doi: 10.1007/978-1-4757-3437-9. URL https://doi.org/10.1007/978-1-4757-3437-9.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, December 2011. ISSN 1537-274X. doi: 10.1198/jasa.2011.tm11181. URL http://dx.doi.org/10.1198/jasa.2011.tm11181.

Ron Elber. Milestoning: An efficient approach for atomically detailed simulations of kinetics in biophysics. *Annual Review of Biophysics*, 49(Volume 49, 2020):69–85, 2020. ISSN 1936-1238. doi: https://doi.org/10.1146/annurev-biophys-121219-081528. URL https://www.annualreviews.org/content/journals/10.1146/annurev-biophys-121219-081528.

Jiahao Fan, Ziyao Li, Eric Alcaide, Guolin Ke, Huaqing Huang, and Weinan E. Accurate conformation sampling via protein structural diffusion. *Journal of Chemical Information and Modeling*, 64(22):8414–8426, 2024. ISSN 1549-960X. doi: 10.1021/acs.jcim.4c00928. URL http://dx.doi.org/10.1021/acs.jcim.4c00928.

Minchao Fang, Chentong Wang, Jungang Shi, Fangbai Lian, Qihan Jin, et al. Halludesign: Protein optimization and de novo design via iterative structure hallucination and sequence design. *bioRxiv*, 2025. doi: 10.1101/2025.11.08.686881. URL https://www.biorxiv.org/content/early/2025/11/09/2025.11.08.686881.

E. Fermi, P Pasta, S Ulam, and M Tsingou. *STUDIES OF THE NONLINEAR PROBLEMS*. May 1955. doi: 10.2172/4376203. URL http://dx.doi.org/10.2172/4376203.

Christopher D. Fu and Jim Pfaendtner. Lifting the curse of dimensionality on enhanced sampling of reaction networks with parallel bias metadynamics. *Journal of Chemical Theory and Computation*, 14(5):2516–2525, Apr 2018. ISSN 1549-9626. doi: 10.1021/acs.jctc.7b01289. URL http://dx.doi.org/10.1021/acs.jctc.7b01289.

Hailong Fu, Gerald Grimsley, J. Martin Scholtz, and C. Nick Pace. Increasing protein stability: Importance of delta c_p and the denatured state. *Protein Science*, 19(5):1044–1052, April 2010. ISSN 1469-896X. doi: 10.1002/pro.381. URL http://dx.doi.org/10.1002/pro.381.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models, 2025. URL https://arxiv.org/abs/2503.00710.

Natali A Gonzalez, Brigitte A Li, and Michelle E McCully. The stability and dynamics of computationally designed proteins. *Protein Engineering, Design and Selection*, 35:gzac001, 2022. doi: 10.1093/protein/gzac001. URL https://doi.org/10.1093/protein/gzac001.

Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. 2023. doi: 10.48550/ARXIV.2305.20009. URL https://arxiv.org/abs/2305.20009.

Donald Hamelberg, John Mongan, and J. Andrew McCammon. Accelerated molecular dynamics: A promising and efficient simulation method for biomolecules. *The Journal of Chemical Physics*, 120(24):11919–11929, 06 2004. ISSN 0021-9606. doi: 10.1063/1.1755656. URL https://doi.org/10.1063/1.1755656.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, et al. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Tatsuki Hori, Yoshitaka Moriwaki, and Ryuichiro Ishitani. Distance-restraint-guided diffusion models for sampling protein conformational changes and ligand dissociation pathways. *Journal of Chemical Theory and Computation*, 22(10):5259–5270, May 2026. ISSN 1549-9626. doi: 10.1021/acs.jctc.6c00199. URL http://dx.doi.org/10.1021/acs.jctc.6c00199.

Zheyuan Hu, Zekun Shi, George Em Karniadakis, and Kenji Kawaguchi. Hutchinson trace estimation for high-dimensional and high-order physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 424:116883, May 2024. ISSN 0045-7825. doi: 10.1016/j.cma.2024.116883. URL http://dx.doi.org/10.1016/j.cma.2024.116883.

John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, Nov 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06728-8. URL https://doi.org/10.1038/s41586-023-06728-8.

Michele Invernizzi and Michele Parrinello. Rethinking metadynamics: From bias potentials to probability distributions. *The Journal of Physical Chemistry Letters*, 11(7):2731–2736, 04 2020. ISSN 1948-7185. doi: 10.1021/acs.jpclett.0c00497. URL https://doi.org/10.1021/acs.jpclett.0c00497.

Giacomo Janson and Michael Feig. Transferable deep generative modeling of intrinsically disordered protein conformations. *PLOS Computational Biology*, 20(5):e1012144, May 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012144. URL http://dx.doi.org/10.1371/journal.pcbi.1012144.

Giacomo Janson, Gilberto Valdes-Garcia, Lim Heo, and Michael Feig. Direct generation of protein conformational ensembles via machine learning. *Nature Communications*, 14(1), February 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36443-x. URL http://dx.doi.org/10.1038/s41467-023-36443-x.

Yaowei Jin, Qi Huang, Ziyang Song, Mingyue Zheng, Dan Teng, and Qian Shi. P2dflow: A protein ensemble generative model with se(3) flow matching. *Journal of Chemical Theory and Computation*, 21(6):3288–3296, 03 2025. ISSN 1549-9618. doi: 10.1021/acs.jctc.4c01620. URL https://doi.org/10.1021/acs.jctc.4c01620.

Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi S. Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. In *ICLR 2023 - Machine Learning for Drug Discovery workshop*, 2023. URL https://openreview.net/forum?id=BgbRVzfQqFp.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. AlphaFold meets flow matching for generating protein ensembles. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22277–22303. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/jing24a.html.

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. In *Advances in Neural Information Processing Systems*, volume 37, pp. 40534–40564. Curran Associates, Inc., 2024b. doi: 10.52202/079017-1282. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/478b06f60662d3cdc1d4f15d4587173a-Paper-Conference.pdf.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL http://dx.doi.org/10.1038/s41586-021-03819-2.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577. Curran Associates, Inc., 2022. doi: 10.52202/068431-1926. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a98846e9d9cc01cfb87eb694d946ce6b-Paper-Conference.pdf.

Helen Lai, Tingyu Wang, Hassan Sirelkhatim, Joe Eaton, Howard Huang, et al. Improving protein-ligand complex generation with force field guidance. *Journal of Cheminformatics*, 18(1), May 2026. ISSN 1758-2946. doi: 10.1186/s13321-026-01198-2. URL http://dx.doi.org/10.1186/s13321-026-01198-2.

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, Sep 2002. ISSN 1091-6490. doi: 10.1073/pnas.202427399. URL http://dx.doi.org/10.1073/pnas.202427399.

Hilbert Yuen In Lam, Sebastian Pujalte Ojeda, Michaela Brezinova, Josef Hanke, Xing Er Ong, Yuguang Mu, and Michele Vendruscolo. Metadiffusion: inference-time meta-energy biasing of biomolecular diffusion models. February 2026. doi: 10.64898/2026.02.10.704873. URL http://dx.doi.org/10.64898/2026.02.10.704873.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761):eadv9817, 2025. doi: 10.1126/science.adv9817. URL https://www.science.org/doi/abs/10.1126/science.adv9817.

Minhuan Li, Jiequn Han, Pilar Cossio, and Luhuan Wu. Robust inference-time steering of protein diffusion models via embedding optimization, 2026. URL https://arxiv.org/abs/2602.05285.

Wen-ran Li, Xavier F. Cadet, David Medina-Ortiz, Mehdi D. Davari, Ramanathan Sowdhamini, Cedric Damour, Yu Li, Alain Miranville, and Frederic Cadet. From thermodynamics to protein design: Diffusion models for biomolecule generation towards autonomous protein engineering, 2025. URL https://arxiv.org/abs/2501.02680.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2, 2025. URL https://openreview.net/forum?id=TZa84ZkOLM.

Yikai Liu, Zongxin Yu, Richard J. Lindsay, Guang Lin, Ming Chen, et al. Exendiff: An experiment-guided diffusion model for protein conformational ensemble generation. *PRX Life*, 3:023013, Jun 2025. doi: 10.1103/PRXLife.3.023013. URL https://link.aps.org/doi/10.1103/PRXLife.3.023013.

Jiarui Lu, Bozitao Zhong, Zuobai Zhang, and Jian Tang. Str2str: A score-based framework for zero-shot protein conformation sampling. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 10663–10694, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/2c3b636b64ca1dfdae3e096e4deeaa42-Paper-Conference.pdf.

Abbas Mammadov, Ozgur Kara, Kaan Oktay, Iskander Azangulov, Adil Kaan Akan, Hyungjin Chung, James Matthew Rehg, and Yee Whye Teh. Exact posterior score estimation for solving linear inverse problems, 2026. URL https://arxiv.org/abs/2606.17048.

Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=YTkQQrqSyE1.

Yinglong Miao, Victoria A. Feher, and J. Andrew McCammon. Gaussian accelerated molecular dynamics: Unconstrained enhanced sampling and free energy calculation. *Journal of Chemical Theory and Computation*, 11(8):3584–3595, 08 2015. ISSN 1549-9618. doi: 10.1021/acs.jctc.5b00436. URL https://doi.org/10.1021/acs.jctc.5b00436.

Ankur Moitra, Andrej Risteski, and Dhruv Rohatgi. The tractability landscape of diffusion alignment: regularization, rewards, and computational primitives, 2026. URL https://arxiv.org/abs/2605.11361.

Milos Musil, Simeon Borko, Joan Planas-Iglesias, David Lacko, Monika Rosinska, et al. Fireprotdb 2.0: large-scale manually curated database of the protein stability data. *Nucleic Acids Research*, 54(D1):D409–D418, November 2025. ISSN 1362-4962. doi: 10.1093/nar/gkaf1211. URL http://dx.doi.org/10.1093/nar/gkaf1211.

Nithin Gopalakrishnan Nair, Anoop Cherian, Suhas Lohit, Ye Wang, Toshiaki Koike-Akino, Vishal M. Patel, and Tim K. Marks. Steered diffusion: A generalized framework for plug-and-play conditional image synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20793–20803. IEEE, October 2023. doi: 10.1109/iccv51070.2023.01906. URL http://dx.doi.org/10.1109/ICCV51070.2023.01906.

Juno Nam, Bálint Máté, Artur P. Toshev, Manasa Kaniselvan, Rafael Gomez-Bombarelli, Ricky T. Q. Chen, Brandon M. Wood, Guan-Horng Liu, and Benjamin Kurt Miller. Enhancing diffusion-based sampling with molecular collective variables. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1bJN1EQByS.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

Jun Ohnuki and Kei-ichi Okazaki. Enhanced sampling of protein conformations in alphafold3 with repulsive bias in the diffusion generative model. *JACS Au*, August 2026. ISSN 2691-3704. doi: 10.1021/jacsau.6c00596. URL http://dx.doi.org/10.1021/jacsau.6c00596.

Alireza Omidi, Jiajun He, Jennifer M. Bui, Jörg Gsponer, and Saifuddin Syed. De novo design of protein switches with diffusion-based ensemble sampling. 2026. doi: 10.64898/2026.07.20.739027. URL http://dx.doi.org/10.64898/2026.07.20.739027.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. 2025. doi: 10.1101/2025.06.14.659707. URL http://dx.doi.org/10.1101/2025.06.14.659707.

Jim Pfaendtner and Massimiliano Bonomi. Efficient sampling of high-dimensional free-energy landscapes with parallel bias metadynamics. *Journal of Chemical Theory and Computation*, 11 (11):5062–5067, Oct 2015. ISSN 1549-9626. doi: 10.1021/acs.jctc.5b00846. URL http://dx.doi.org/10.1021/acs.jctc.5b00846.

Ieva Pudžiuvelytė, Kliment Olechnovič, Egle Godliauskaite, Kristupas Sermokas, Tomas Urbaitis, et al. Temstapro: protein thermostability prediction using sequence representations from protein language models. *Bioinformatics*, 40(4), March 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae157. URL http://dx.doi.org/10.1093/bioinformatics/btae157.

Rishwanth Raghu, Axel Levy, Gordon Wetzstein, and Ellen D Zhong. Multiscale guidance of protein structure prediction with heterogeneous cryo-EM data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=T3DF4yxcPz.

Daniel D. Richman, Jessica Karaguesian, Carl-Mikael Suomivuori, and Ron O. Dror. Unlocking hidden biomolecular conformational landscapes in diffusion models at inference time, 2025. URL https://arxiv.org/abs/2512.03312.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Jp988ELppQ.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, et al. BoltzGen: Toward universal binder design. *bioRxiv*, 2025. doi: 10.1101/2025.11.20.689494.

Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, 314(1):141–151, 1999. ISSN 0009-2614. doi: https://doi.org/10.1016/S0009-2614(99)01123-9. URL https://www.sciencedirect.com/science/article/pii/S0009261499011239.

G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, Feb 1977. ISSN 0021-9991. doi: 10.1016/0021-9991(77)90121-8. URL http://dx.doi.org/10.1016/0021-9991(77)90121-8.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, et al. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60515–60529. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/uehara25a.html.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 160217–160252. Curran Associates, Inc., 2025. doi: 10.52202/085713-5354. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/eacce1f7d11e2c3a7568468e0eff5d33-Paper-Conference.pdf.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, Aug 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 democratizing biomolecular interaction modeling. November 2024. doi: 10.1101/2024.11.19.624167. URL http://dx.doi.org/10.1101/2024.11.19.624167.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 31372–31403. Curran Associates, Inc., 2023. doi: 10.52202/075280-1363. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/63e8bc7bbf1cfea36d1d1b6538aecce5-Paper-Conference.pdf.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, et al. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40001–40039. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/yim23a.html.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23117–23127, 2023. doi: 10.1109/ICCV51070.2023.02118.

Min Zhao, Fan Bao, Chongxuan LI, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3609–3623. Curran Associates, Inc., 2022. doi: 10.52202/068431-0261. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/177d68f4adef163b7b123b5c5adb3c60-Paper-Conference.pdf.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, et al. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00837-3. URL https://doi.org/10.1038/s42256-024-00837-3.

Daniel M. Zuckerman and Lillian T. Chong. Weighted ensemble simulation: Review of methodology, applications, and software. *Annual Review of Biophysics*, 46(Volume 46, 2017):43–57, 2017. ISSN 1936-1238. doi: https://doi.org/10.1146/annurev-biophys-070816-033834. URL https://www.annualreviews.org/content/journals/10.1146/annurev-biophys-070816-033834.

A Analytical estimates for guidance

A.1 Cumulant expansion for guidance

For a generic scalar function u and random vector x, we use

$$\ln\left\langle e^{u(x)}\right\rangle \approx u(\langle x\rangle)+\frac{1}{2}\operatorname{Tr}\left[H_u(\langle x\rangle)\mathrm{Cov}(x)\right]+\frac{1}{2}\boldsymbol{\nabla}u(\langle x\rangle)^T\mathrm{Cov}(x)\boldsymbol{\nabla}u(\langle x\rangle)+\ldots, \tag{6}$$

where $(H_u)_{ij}(x)=\partial_{ij}u(x)$ is the Hessian and $\mathrm{Cov}(x)=\left\langle xx^T\right\rangle-\langle x\rangle\langle x\rangle^T$.

Using the denoiser model to approximate $\mathbb{E}_{x_0|t,x,c}[x_0]\approx D_\theta(t,x,c)$,

and gaussian noising $X_t=X_0+\epsilon_t\xi$ with $\xi\sim\mathcal{N}(0,1)$ we get

Tweedie/Stein (Efron, 2011) $\boldsymbol{\nabla}_x\ln p(t,x,c)=(\mathbb{E}_{x_0|t,x,c}[x_0]-x)/\epsilon_t^2$

and a similarly-obtained (Meng et al., 2021) $\boldsymbol{\nabla}_x\boldsymbol{\nabla}_x^T\ln p(t,x,c)=(\mathrm{Cov}_{x_0|t,x,c}[x_0]-\epsilon_t^2\mathbb{I})/\epsilon_t^4$

So $\mathrm{Cov}_{x_0|t,x,c}[x_0]\approx e_t^2\boldsymbol{\nabla}_xD_\theta^T(t,x,c)$.

Only the symmetric part contributes to guidance, so a possible slight asymmetry due to approximation $\mathbb{E}\approx D_\theta$ is not a problem. The $\succeq$ also might be violated, but not significantly given the denoiser is well trained.

This gives 3 approximation levels for d_{guided}, which balance accuracy with computational cost:

Using $m=m(x)=D_\theta(t,x,c)\in\mathbb{R}^d$, where $d\sim10^{4-5}$ for proteins,

$J=\boldsymbol{\nabla}_xm^T(x)\in\mathbb{R}^{d\times d}$, so $J_{ij}=\partial_jm_i$,

$g=(\boldsymbol{\nabla}u_b)(m)\in\mathbb{R}^d$,

$H=(\boldsymbol{\nabla}\boldsymbol{\nabla}^Tu_b)(m)\in\mathbb{R}^{d\times d}$,

$r_k=\mathrm{Tr}\left[(\partial_kH)J^T\right]=\sum_{i,j}J_{ij}\partial_kH_{ij}$,

$s_k=\sum_{i,j}(g_ig_j-H_{ij})(\partial_kJ_{ij})$

1st order (DPS-like guidance)

$$d_{guide}(t,x,c)-d_{bg}(t,x,c)=d_{corr}(t,x,c)\approx e_t\boldsymbol{\nabla}_xu_b(m(x))=e_tJ^Tg \tag{7}$$

where the $\boldsymbol{\nabla}_xu_b(m(x))$ is used in practice to take advantage of autodiff and VJPs to avoid constructing J.

1.5 order (still fully VJP-compatible, with some information from the 2nd order, but not all)

$$d_{corr}(t,x,c)\approx e_tJ^T\left(\mathbb{I}-e_t^2HJ^T\right)g \tag{8}$$

which also can be represented by VJPs + grad-stopping, thus avoiding constructing J or H explicitly.

2nd order (need explicit J and H, but full 2nd order)

$$d_{corr}(t,x,c)\approx e_t\left[J^T\left(\mathbb{I}-e_t^2HJ^T\right)g+\frac{e_t^2}{2}(J^Tr-s)\right]=e_t\boldsymbol{\nabla}_x\left[u(m)+\frac{e_t^2}{2}(\langle H,J\rangle_F+g^TJg)\right] \tag{9}$$

This requires explicit J and H construction, so may require more resources. This can be avoided at a cost of approximating the $\langle H,J\rangle_F$ via Hutchinson trace estimator (Hu et al., 2024), which is asymptotically exact.

A.2 Noise scaling of guidance corrections

For a simple end-to-end stretching/compactification bias $u_b(X)=k_0(x_N-x_C)^2/2R_g^2$, of strength k_0, between the N- and C-terminal atom positions x_N,x_C of a chain with radius of gyration R_g, we

Table 2: Scaling of guidance terms

	$\lvert d_{bg}\rvert$	$\lvert$1-o$\rvert$	$\lvert$1-o$\rvert/\lvert d_{bg}\rvert$	$\lvert e_t^2 H J^T\rvert \sim \lvert$1.5-o/1-o $-1\rvert$
$\xi_t \gg 1$	1.0	$k_0/\xi_t \ll k_0$	$k_0/\xi_t \ll k_0$	k_0
$\xi_t \sim 1$	1.0	k_0	k_0	k_0
$\xi_t \ll 1$	$\xi_t \ll 1$	$k_0\xi_t \ll k_0$	k_0	$k_0\xi_t^2 \ll k_0$

can estimate how the magnitudes of the 1-o and 1.5-o guidance terms scale with the noise level, in terms of the dimensionless ratio $\xi_t \equiv e_t/R_g$.

Plugging a gaussian $p(t,x)$ for estimates, now to $D(t,x) \approx x + \epsilon_t^2 \boldsymbol{\nabla}_x \ln p(t,x) \sim x - \epsilon_t^2 x/(R_g^2 + \epsilon_t^2) = x/(1+(\epsilon_t/R_g)^2)$. So, $|J| \sim 1/(1+(\epsilon_t/R_g)^2)$. This makes sense, since for small noise $\epsilon_t/R_g \ll 1$ we get $|J| \sim 1$, so near-identical transform, and for large noise $\epsilon_t/R_g \gg 1$, we get $|J| \sim (R_g/\epsilon_t)^2 \ll 1$.

Thus, for $|d_{guide}| \sim \epsilon_t |J| |\boldsymbol{\nabla} u_b|$ with $|\boldsymbol{\nabla} u_b| \sim k_b |r_1 - r_{-1}| \sim k_b R_g \sim k_0/R_g$, we get $|d_{guide}| \sim \epsilon_t/(1+(\epsilon_t/R_g)^2) \cdot k_0/R_g = k_0/(\epsilon_t/R_g + R_g/\epsilon_t)$. So, for both $\epsilon_t \gg R_g$ and $\epsilon_t \ll R_g$, we expect $|d_{guide}| \ll k_0$. For $\epsilon_t/R_g \sim 1$, we get $|d_{guide}| \sim k_0$. $|d_{guide}| \sim k_0/(\epsilon_t/R_g + R_g/\epsilon_t)$.

Then we can compare $|d_{guide}|/|d_{BG}| \sim k_0/\sqrt{1+(\epsilon_t/R_g)^2}$. The 1.5 order correction, given $|H| \sim k_0/R_g^2$, is $|\epsilon_t^2 H J^T| \sim k_0/(1+(R_g/\epsilon_t)^2)$, which is subtracted from $\mathbb{I}$, so should be compared to 1.0 always.

So, for high noise $\epsilon_t \gg R_g$ we get $|d_{guide}|/|d_{BG}| \sim k_0 \cdot R_g/\epsilon_t \ll k_0$, and for low noise $|d_{guide}|/|d_{BG}| \sim k_0$, where k_0 is a typical interaction strengths at a typical distance $k_0 \sim |u_b(R_g)|$.

These estimates are later confirmed on a 2D toy model (Section 4.1). At large noise ($\xi_t \gg 1$), the first-order correction is a small fraction of the base drive and barely perturbs sampling; at $\xi_t \sim 1$, the correction is comparable to the base drive and most strongly steers the trajectory; at small noise ($\xi_t \ll 1$), the correction shrinks in absolute size and the 1.5-order curvature term becomes negligible.

Analysis of the 2nd order can be done in a similar way. One can do "perturbation theory" of the evolving distribution, e.g. starting from

$$\ln p(t,x) \approx -\frac{1}{2}\left(\frac{x}{R_g(t)}\right)^2 - a\left(\frac{x}{R_g(t)}\right)^4 + \mathcal{O}(\ln R_g(t)) \tag{10}$$

because the term s and the 2nd order correction are 0 for a pure gaussian. A better way is to use Hermite bias to write the data distribution as

$$p_0(x) = \phi_{R_g}^{(d)}(x)\left[1 - \sum_{|\boldsymbol{n}|\geq 1} c_{\boldsymbol{n},0} H_{\boldsymbol{n}}(x/R_g)\right] \tag{11}$$

and use a property

$$\left[\phi_{R_g}^{(d)} H_{\boldsymbol{n}}(\cdot/R_g)\right] * \phi_{\epsilon_t}^{(d)} = \rho_t^{|\boldsymbol{n}|} \phi_{R_t}^{(d)} H_{\boldsymbol{n}}(\cdot/R_t), \qquad \rho_t := \frac{R_g}{R_t}. \tag{12}$$

to get $p(t,x_t)$ and thus scaling of its corrections. In the end, this more rigorous treatment with non-zero H_3 and H_4 terms gives the same scaling as we estimated.

B 2D TOY-MODEL DETAILS

B.1 SETUP

$$u_{2D}(x) = \left(\frac{x}{s}\right)^2 - \sum_i m_i \exp\left[a_i(x-x_i)^2 + b_i(x-x_i)(y-y_i) + c_i(y-y_i)^2\right], \tag{13}$$

with $s = 3$, $a = (-1.0, -1.0, -6.5)$, $b = (0.0, 0.0, 11.0)$, and $c = (-10.0, -10.0, -6.5)$; the depths m_i are varied to control basin separation.

B.2 ARCHITECTURE

The toy diffusion model is a simplified and miniaturized variation of BoltzGen's diffusion module (Stark et al., 2025) that keeps the diffusion formulation, sampler and guidance code path unchanged and replaces only the network.

As in BoltzGen, the denoiser uses EDM preconditioning (Karras et al., 2022), $D_\theta(x, \sigma) = c_{skip}x + c_{out}F_\theta(c_{in}x,\, c_{noise})$ with $c_{skip} = \sigma_{data}^2/(\sigma^2+\sigma_{data}^2)$, $c_{out} = \sigma\sigma_{data}/\sqrt{\sigma^2+\sigma_{data}^2}$, $c_{in} = 1/\sqrt{\sigma^2+\sigma_{data}^2}$ and $c_{noise} = \frac{1}{4}\ln(\sigma/\sigma_{data})$. It is trained on the loss $\lambda(\sigma)\|D_\theta(x_0+\sigma\epsilon, \sigma) - x_0\|^2$ with $\lambda = (\sigma^2+\sigma_{data}^2)/(\sigma\sigma_{data})^2$ and $\ln(\sigma/\sigma_{data}) \sim \mathcal{N}(-1.2,\, 1.5^2)$. Sampling uses the AF3 schedule (Abramson et al., 2024) ($\sigma \in [4{\cdot}10^{-4}, 160]\,\sigma_{data}$, $\rho = 7$, 100 steps), the same stochastic churn ($\gamma_0 = 0.8$) and noise scale (1.003), and an Euler update. The guidance correction d_{corr} (equation 4 or its higher orders, Section A) is added at the same point of each step. RipGuide samples are initialized by partially noising X_{init}, as for proteins.

In BoltzGen, F_θ consists of three parts: an atom-attention encoder (3 layers, 4 heads) that pools atoms into tokens; a 24-layer, 16-head token-level DiffusionTransformer of width 768; and an atom-attention decoder (3 layers) that outputs per-atom coordinate updates. Every layer is conditioned on the noise level via adaptive LayerNorm and sigmoid gates, driven by a fixed random Fourier embedding of c_{noise} combined with the trunk single representation. Attention is biased by pair representations from a Pairformer trunk that is evaluated once per design.

A single 2D point has no atoms, tokens, pairs or conditioning inputs, so attention and the trunk have nothing to act on. We therefore replace F_θ by a residual MLP that keeps BoltzGen's noise-conditioning scheme. c_{noise} is embedded as $\cos(2\pi(w\, c_{noise} + b))$ with fixed random $w, b \in \mathbb{R}^{32}$, followed by LayerNorm, a linear layer and a SiLU. The input $c_{in}x \in \mathbb{R}^2$ is projected to width 8 and passed through 3 residual blocks. Each block applies a LayerNorm with shift and scale predicted from the noise embedding, then two SiLU-activated linear layers, and adds the result back through a gate that is also predicted from the embedding. A final LayerNorm and linear layer output $r_{update} \in \mathbb{R}^2$. The gate and output projections are zero-initialized, so $D_\theta = c_{skip}x$ at initialization, as in BoltzGen. The model has 3,986 trainable parameters, $\sim 10^5$ times fewer than BoltzGen's F_θ.

The remaining deviations from BoltzGen are as follows. First, $\sigma_{data} = 1$ instead of 16 Å, matching the $\mathcal{O}(1)$ scale of u_{2D}. Second, there is no centering, random SE(3) augmentation, rigid alignment, smooth-lDDT term or atom-type loss weighting: these serve only the rotational invariance and chemistry of atomic point clouds, and centering would collapse a single point to the origin. Third, the step scale is 1 (the exact EDM Euler step), because the over-relaxed step scales used by BoltzGen (> 1) concentrate low-dimensional samples onto the data mean. Fourth, each of the two coordinates is treated as an independent bias degree of freedom, the analogue of a native-contact distance r_{ij}: the $RMSD$-family collective variables are softened per coordinate and then averaged, exactly as over native contacts for proteins.

The network was trained on $5{\cdot}10^5$ i.i.d. samples from $p \propto e^{-u_{2D}}$ (80/20 train/test split), drawn by inverse-CDF sampling on an 801×801 grid, using Adam (learning rate $3{\cdot}10^{-3}$, batch size 16384, 5000 epochs) and an EMA of the weights (decay 0.999). Because the network is tiny and smooth, the exact second-order guidance correction, which requires third derivatives of u_b and second derivatives of D_θ, is tractable by nested automatic differentiation, which is not feasible at BoltzGen's scale.

B.3 ANALYSIS

To test the necessity of the bias, we kept the dynamical scheme $e_{noise,n} = \alpha \left\langle RMSD_p^{(k)} \right\rangle_{k\in[-N_{top}:]}$, but disabled the repulsive bias accumulation. Different values of α, mean-implementations and $RMSD$-flavors were tested. Results for the 2D toy-model in the main text use $\alpha = 1.0$, $N_{top} = 10$, $RMSD = RMSD_{p=2}$, $\langle . \rangle = (\sum_k^{N_{top}} RMSD^{(k)}/N_{top})^{1/p}$. We tested both $e_{init} \ll W_{well}$ (figure 5a) and $e_{init} \sim W_{well}$ (figure 5b). Small e_{init} with the dynamical scheme active is not enough to increase e_{noise} to discover other basins. Sufficient $e_{init} \geq W_{well}$ starts sampling all the reachable basins right away, so they are discovered, however, there is no notion of which ones are closer. This is problematic in 2 ways. First, if a too-large e_{init} is chosen that reaches some wells that are deeper than the original well but more distant than "disturbed" states of the same well - sampling will be mostly of these deeper basins, not of the less-stable but near-by states. This is less representative of "failure modes" of a protein and is closer to producing significantly different conformations of the same proteins. Second, if e_{init} again is not chosen well, many distant states may be sampled before closer "unfolding" states, which makes it difficult to determine when to label sampling complete, at least based on the "too large RMSD-s" criteria, which is currently used. For a given system, it should be possible to tune e_{init} so that it reaches only the closest next basin, thus the dynamical scheme will increase e_{noise} once this next basin is discovered, but not before. This will produce gradual discovery without the bias. However, this requires correctly setting (guessing) e_{init} with accuracy that makes this system-specific. Instead, starting with very small e_{init} and using the bias to help the system escape the original basin is an approach that finds the right e_{noise} for all investigated systems regardless of the exact initial e_{init}.

Another requirement to discover wells in the right order is that the wells need to be well-defined. Wells of the depth $dU \sim 2-3kT$ were found not sufficient for meaningful discovery (figure 6-top), while wells with $dU \sim 4-5kT$ were already sufficient. Perhaps the threshold of 2-3kT can be improved to slightly lower values, but this points out an expected constraint that since sampling is done in discrete number of trials (i.e. we cannot sample 1/10-th of a sample), then the sampling cannot be slowed-down indefinitely so wells that are shallower than a few kT will be insufficient to bias sampling to them as opposed to wider regions of phase-space that might be slightly higher in energy but are much wider, and thus have more entropic pull. All the above are just observational estimates since our simple DPS-like guidance does not guarantee to skew the sampled distribution exactly by $\sim e^{-u_b(x)}$.

Real protein states are high-dimensional, so visualizing their ensembles for basin-identification is difficult. Thus, we need a unified algorithm for labeling states as being already outside of the original basin but not yet blown-up. We use a simple $RMSD_p^*$-based separation, which works for the 2D toy model figure 9.

The general metaD-like idea that basins will be gradually filled before the e_{noise} can increase is supported by the fact that filling-up with weaker gaussians (smaller a_{metaD}) allows to discover shallower basins before "blowing-up" (i.e. starting frequently sampling the wider space), as shown in figure 6.

An interesting observation is that biasing with the 1-o scheme seems to require more flattening of the landscape in order to discover new basins than with the 1.5-o scheme: In figure 7, 1-o flattened-out the systems to $\sim 0.5kT$ wells before blowing-up, and thus was able to catch the shallowest basin at (-0.5,1), while 1.5-o only flattened the landscape to $\sim 2kT$ wells before blowing-up. This resulted in only 1 point in the shallowest basin, which would be practically indistinguishable from random sampling without a known answer. However, the blow-up (which is treated as convergence of all basins below the general flat landscape level) was achieved with the same bias strength $\sim 3-4$ times faster, which may be advantageous for exploring landscapes that are expected to be more rough but don't require such exhaustive investigation.

The reason why 1.5-o scheme seems to provide more disturbance given the same bias magnitude may be explained by the scalings of different guidance terms with noise, estimations of which were summarized above and detailed in SI. figure 8 shows the predictions for both 1-o and 1.5-o terms scaling match the numerical results for both big and small noise limits. However, for intermediate noise, where guidance seems to be most impactful, the relative contribution of the terms is 1-o/1.5-

o$\sim |H_u|R_g^2$, so, for $u(x)$ with details σ_d finer than $R_g = \sigma_{data}$ scale, this is $\sim (R_g/\sigma_d)^2 > 1$ and thus $1.5 - o$ correction is stronger than 1-o.

One important difference between this 2D model and real protein is that real proteins have many more DOFs. A known metaD-style approach to high-dim problems is so-called parallel-bias metaD (PBmetaD [..]) that applies bias alongs many 1D CVs instead of the full nD scape. figure 9 shows that applying this in our 2D case works well for gradual discovery of basins and also follows the derived scalings of terms contributions. The powers in RMSD are found to matter more and the simplest choice of (2,2) is found to work better for PBmetaD. Changing the bias strength below dU_{well} does not seem to significantly affect how gradual the discovery is (figure 10) until the overall sampling becomes too fast before the blow-up for $a_{metaD} > dU_{wells}$ so that shallower wells do not get sufficient sampling. Peaks with $\sigma_d \sim W_{well}$ are able to blow-up the system eventually, but break the real order of discovery and also produce many false-discovery points due to strong geometrical artifacts from PB-nature of sampling (figure 11-top). Using $\sigma_d > W_{well}$ results in the bias being nearly-constant in the well, which eliminates its effect and does not lead to any changes in dynamics from the no-bias case. Finally, using $e_{init} \geq W_{well}$ with bias, similar to the no-bias case, samples all basins, but does not give the gradual nature of discovery (figure 12).

C RIPGUIDE: DETAILS OF DESTABILIZATION

C.1 THE FULL RIPGUIDE ALGORITHM

Algorithm 2: Weak-link identification via gradual destabilization. "Q[U]" means Q is in units U. X_{orig}[A]: 3D coordinates of all atoms in the original structure; σ_{init}[A]: magnitude of initial noising; $RMSD_{max}$[A]: deviations that are considered unnaturally big (blow-ups); σ_d[A]: typical range of repelling potential; a_{metaD}[1]: repelling bias strength; p_{blowup}[1]: power that allows to emphasize larger deviations; dI_{min}[1]: minimum distance in AA-index space for atoms to be considered for being native-contacts of each-other; $R_{cut,NC}$[A]: maximum distance in 3D space for atoms to be considered for being in native-contact; N_{last}[1]: the number of "latest" samples considered for blow-up criteria and for latest-noise level determination; $\alpha_{rescale}$[1]: a scaling factor between how large latest deviations are and how large the next noising will be; p_{noise}[1]: a power exponent that tunes emphasis whether maximum-vs-average deviation among the latest ones determine the next noising magnitude; blow_freq$_{max}$[1]: fraction of blown-up samples in the latest samples that indicates convergence of destabilization in a sense that basin near X_{orig} are considered filled-up; $R_{cut,clust}$[A]: cut-off distance in 3D space for clustering of atom-disturbances; weak_atoms_frac[1]: the approximate fraction of all atoms that we desire to see labeled into weak-link regions;

Require: $X_{orig} \in \mathbb{R}^{N_{atoms}\times 3}$, $\sigma_{init} > 0$, $RMSD_{max} > 0$, $\sigma_d > 0$, $a_{metaD} > 0$, $p_{blowup} > 0$, $dI_{min} \in \mathbb{N}_+$, $R_{cut,NC} > 0$, $N_{last} \in \mathbb{N}_+$, $\alpha_{rescale} > 0$, $p_{noise} > 0$, $1 >$ blow_freq$_{max} > 0$, $R_{cut,clust} > 0$, $1 >$ weak_atoms_frac > 0

$N_{AA} \leftarrow N_{atoms}/14$; `// BoltzGen has 14 pseudo-atoms per amino-acid`
design_mask$[0 : N_{AA}] \leftarrow False$;
$NC \leftarrow$ NativeContacts$(X_{orig}, dI_{min}, R_{cut,NC})$;
$n \leftarrow 0$;
$X_0 \leftarrow X_{orig}$;
$\sigma_{noise}[1] \leftarrow \sigma_{init}$;
blow_up $\leftarrow False$;
while *not blow_up* **do**
  $u_n(X, NC, \sigma_d, X_{0\dots n}) \leftarrow a_{metaD} \sum_{k=0}^{n} u_{repel}(X, NC, \sigma_d, X_k)$; `// Accumulate repelling bias`
  $n \leftarrow n + 1$; `// Just a counter`
  $X_n^{noised} \leftarrow X_{orig} + \sigma_{noise}[n] \cdot \mathcal{N}(0, 1)$; `// Noise the original structure to a degree comparable to Nlast latest deviations`
  $X_n \leftarrow$ RipGuide$(X_t = X_n^{noised}, \sigma_t = \sigma_{noise}[n], u_b(x) = u_n(x, \dots),$ design_mask$)$; `// Denoise under bias`
  dists_blowup$[n] \leftarrow RMSD^*_{p_{blowup}}(X_{orig}, X_n, NC)$;
  dists_noise$[n] \leftarrow RMSD^*_{p_{noise}}(X_{orig}, X_n, NC)$;
  if $n > N_{last}$ **then**
    blow_up $\leftarrow$ mean(dists_blowup$[-N_{last} :] > RMSD_{max}) >$ blow_freq$_{max}$; `// Often blow-ups`
    $\sigma_{noise}[n+1] \leftarrow \alpha_{rescale} \cdot$ mean(dists_noise$[-N_{last} :]^{**} p_{noise})^{**}(1/p_{noise})$; `// Noise up to last deviations`
  else
    $\sigma_{noise}[n+1] \leftarrow \sigma_{noise}[n]$;
  end
end
$N_{structures} \leftarrow n$;
$I_{optimal} \leftarrow \{k \in [1; N_{structures}] :$ dists$[k] < RMSD_{max}\}$; `// Not too-broken structures`
$\forall i \in [1; N_{atoms}] : w_i = \frac{1}{|I_{optimal}|} \sum_{k \in I_{optimal}} \sum_{j \in NC[i]} \left(\frac{|r_{k,ij} - r_{1,ij}|}{\sigma_d} \right)^p$; `// How disturb-able each atom is`
$w_{thr} \leftarrow$ argmin$_v$ $(|\|$DBSCAN$(X_{orig}, R_{cut,clust}, N_{min} = v,$ weights $= \{w\})\| - N_{atoms} \cdot$ weak_atoms_frac$|)$; `// Find Nmin for DBSCAN that gives approximately desired number of weak-link atoms`
weak_link_regions $\leftarrow$ DBSCAN$(X_{orig}, R_{cut,clust}, N_{min} = w_{thr},$ weights $= \{w\})$; `// Final result: sets of atom-indices of weak-link regions`

where

$$NC(X, dI_{min}, R_{cut}) = \{(i,j) \in \text{Atoms}(X) : |I_{AA}(i) - I_{AA}(j)| > dI_{min} \wedge r_{ij} \leq R_{cut}\} \quad (14)$$

$$u_{repel}(X, NC, \sigma_d, X_1) = \sum_{(i,j)\in NC} \exp\left[-\left(\frac{r_{ij} - r_{1,ij}}{\sigma_d}\right)^2\right] \quad (15)$$

$$RMSD^*_p(X, X_1, NC) = \left(\frac{1}{\|NC\|}\sum_{(i,j)\in NC} |r_{ij} - r_{1,ij}|^p\right)^{1/p} \quad (16)$$

Values we found to work well (and thus used, unless specified otherwise) for real proteins: $\sigma_{init} = 0.01$[A], anything below a native fluctuation level of $\sim 0.1 - 0.2$A behaves similarly; $RMSD_{max} = 6$[A], can be increased to ~ 10A; $\sigma_d = 1.5$[A], values in $\sim 1.0 - 2.0$ seem optimal; $a_{metaD} = 5$, range $\sim 5 - 10$ seems optimal, lower values are possible for more gentle exploration; $p_{blowup} = 5$, but no qualitative difference in results between 2-5; $dI_{min} = 2$, motivated by biology (α-helix structure), so have not tried different; $R_{cut,NC} = 4$[A], range $3.5 - 4.5$ seems optimal; $N_{last} = 10$, no qualitative difference in 3-10 range; $\alpha_{rescale} = 1.5$, range 1.2-2.0 seems optimal, lower values need significantly longer destabilization runs, higher values exploit occasional fluctuations too aggressively; $p_{noise} = 5$, but no qualitative changes in the range 2-5; blow_freq$_{max} = 0.5$, no qualitative change in 0.2-0.8 range; $R_{cut,clust} = 4$[A], 3.5-4.5A seems optimal; weak_atoms_frac $= 0.1$, values below 0.05 return results too sensitive to fluctuations, above 0.15 includes too vast regions.

C.2 Destabilization reproducibility

Per-atom PN-scores w correspond to protein physical structure: 10 statistically independent destabilization RipGuide runs were conducted for 4 proteins. Figure 13 quantifies the reproducibility of PN-scores$= w$ by showing the mean and variance of w-s of each atom from the corresponding 10 runs. The variation of w-s between atoms are considerably higher than the variance of w-s of a given atom between statistically independent runs, indicating reproducibility. This is quantified by the noise-to-signal ratio defined as $\text{mean}_{\text{atom-ID}}(\text{std}_{\text{random-seed}}[\ln(w)])/\text{std}_{\text{atom-ID}}(\text{mean}_{\text{random-seed}}[\ln(w)])$ and it takes values of 0.21, 0.18, 0.45, 0.44. The similarity matrices in Figure 15 show how much each pair of weak-link regions (sets of AAs) identified in 10 independent runs overlap.

PN-scores also show correspondence to the protein structure. Taking Figure 13a as an example, two kinds of structures are visible: First, the higher "frequency" oscillations with the period of ~ 29 atoms, which, assuming 8 heavy atoms per AA, means ~ 3.625 AAs, which well matches the α-helix known structure. This corresponds well with the α-helical structure. This works as intended since if an α-helix does break-off, the patching should focus on the regions that lose more native contacts in the process of this breakoff, which is precisely the internal α-helices' regions in case of Figure 4a. Second, 4 larger parts are visible in Figure 13a, separated by atom-indices $\sim 200, 400, 570$. These correspond to 4 α-helices of the protein.

Par-AA labeling of weak-link regions is reproducible: Once the per-atom PN-scores are calculated, the weak-link regions are obtained by clustering atoms via DBSCAN with weights=PN-scores, $R_{cut} = 4$ Åand N_{min} chosen to have approximately 10% of atoms in the weak-link regions. The same 10 RipGuide runs for each of 4 proteins yield 10 sets of weak-link regions for each proteins. For each protein, each pair among these 10 regions often includes same AAs, which allows us to define the similarity matrix $\text{similar}_{ij} = |\text{overlap}_{ij}|/|\text{union}_{ij}|$, which gives $(61 \pm 15)\%$, $(91 \pm 8)\%$, $(67 \pm 16)\%$ and $(66 \pm 10)\%$ correspondingly (Figure 15).

PN-scores give extra information compared to pLDDT. A negative correlation between PN-scores and pLDDT is expected since high pLDDT often means high order, but high order is generally harder to break, which should yield lower PN-scores if RipGuide works as intended. We indeed see (Figure 14) a weak negative correlation $\sim -(0.15 - 0.3)$ for 3 systems with lower reproducibility of weak-link regions (~ 0.65) and correlation of $\sim -(0.5 - 0.7)$ for the rucaparib case, which had the highest reproducibility of the weak-link regions (~ 0.9). These numbers imply that PN-scores

behave in a generally expected way, but provide information complementary of pLDDT. The few single-atom outliers with very low PN-scores are atoms with very few native contacts (usually on the periphery of the protein structure), so they do not even have a chance of gaining a high PN-score. Their existence does not affect downstream clustering since only high-PN atoms are included in weak-link regions.

C.3 More discussion of RipGuide

The mechanisms of thermal destabilization and RipGuide are somewhat different: physical heating with $\dot{T} > 0$ decreases the barriers as $\Delta u^{\dagger}(t) \sim \Delta U^{\dagger}/T(t)$ so that $\Delta \dot{u}^{\dagger}(t) < 0$, while our bias-accumulation approach with $\dot{a}_{bias} > 0$ does it as $\Delta u^{\dagger}(t) \sim \Delta U^{\dagger}/T - a_{bias}(t)$ so again $\Delta \dot{u}^{\dagger}(t) < 0$. RipGuide also does not account for real-time dynamics that can cause the real system to take a path different from the vicinity of the minimal free-energy path. However, inertial movement typically decays faster than unfolding, thus the quasi-equilibrium assumption is useful for many proteins.

The sequence of both binder and the target are fully frozen during RipGuide. However, no pair-distance conditioning is applied, so the target is not forced to preserve its conformation. This is another indirect evidence that destabilization is caused by the RipGuide bias, not only the dynamically increased noise since the target, which is noised to the same level before denoising but does not have a RipGuide bias applied, preserves its structure unlike the binder that breaks.

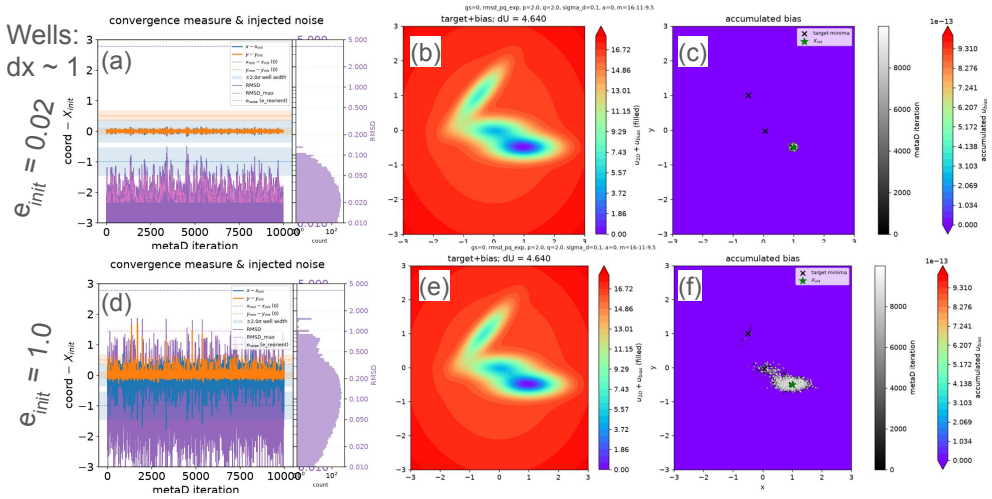

Figure 5: ”simulation” (i.e. sequential sampling) of states without bias ($a_{metaD} = 0$) but still with active dynamical scheme for adjusting noising level of new samples according to . The typical well width W_{well} 1 (labeled dx here) is given for reference. Since the dynamical scheme determines e_{noise} for all steps after the 1st one, the only control is e_{init}. (top) Small noise $e_{init} \ll W_{well}$: no discovery of new basins. (right) Sufficient noise $e_{init} \sim W_{well}$: discovery of all basins roughly within e_{init} occurs, however, the approach does not return the correct order of discovery that physically should be related to how close basins are to the initial point, which is at (1,-0.5). Finally, sampling basins in this no-bias formulation requires first initiating the denoising not too far from a basin and then doing it enough times to naturally randomly rarely draw from higher-energy basins. Even the first step of nearly-finding a basin is exponentially hard in high-dim cases like real proteins.

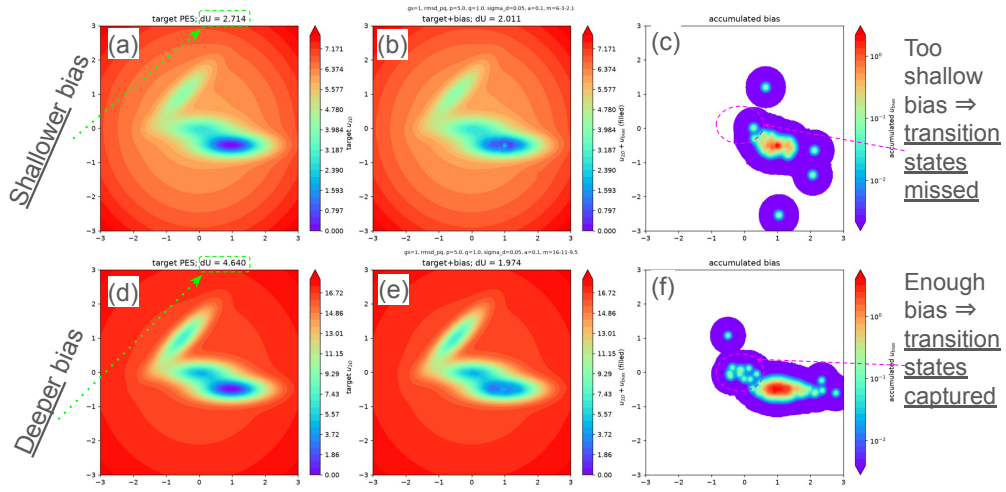

Figure 6: Wells with depth $\leq 2 - 3kT$, caused by either initial depth of the bias potential, are well sampled in random order without help of any bias. This is expected since it should take $\sim e^{\Delta u}$ attempts to sample a point at Δu up from the minimum value of u inside the region $R \sim e_{noise}$. (top) Well-depths are not sufficient to significantly bias sampling to them, so the general outer space starts being sampled after the original well if filled-up. (bottom) Deeper wells $\geq 4 - 5kT$ are enough to meaningfully bias sampling to them, allowing the discovery of the next basin after the original basin fills up.

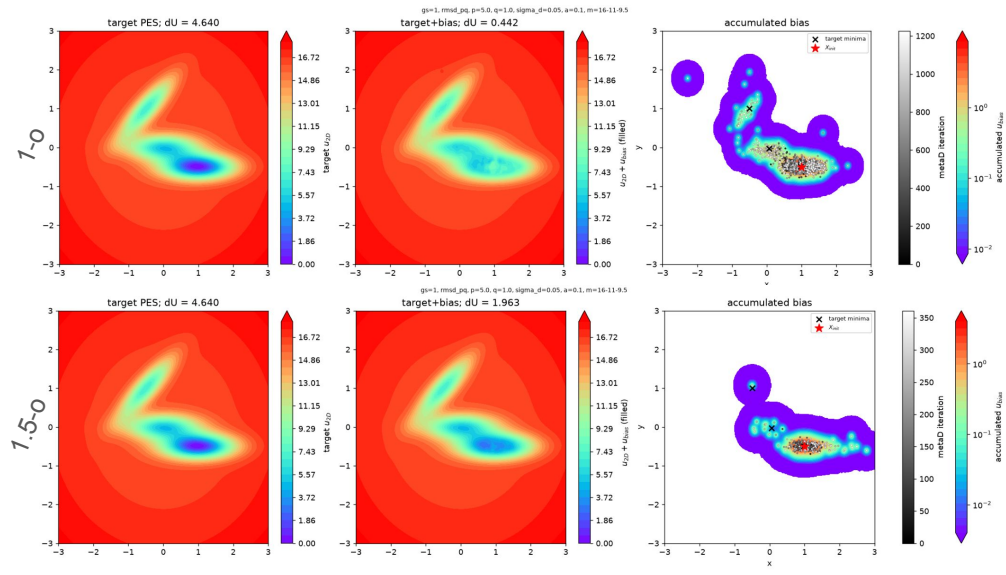

Figure 7: The difference between 1-o and 1.5-o guidance schemes at the same bias strength. (top) The usual DPS "1st-order" guidance. Seems less prone to blow-up, which allows sampling shallower wells. (bottom) Seems more prone to blow-up and overcoming barriers.

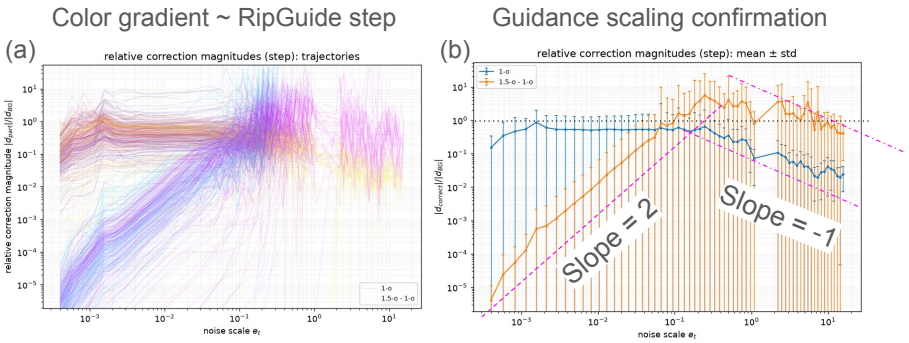

Figure 8: (a) Each denoised point is a trajectory of the magnitude of a correction relative to the base-model drive d_{BG}. Trajectories start at $e_t = e_{noise}$ and go to lower noise down to $s_{min} = 4 \cdot 10^{-4}$. Color gradients within trajectory-clouds correspond to the "time" (metaD step) that passed when this trajectory was started. Dark-blue→yellow is for the 1-o term, light-blue→pink is the addition of 1.5-o relative to 1-o. Interestingly, the strength of the drive does not grow with bias accumulation, which indicates the new points not only end-up in mostly unsampled regions, but also follow paths that are trying to avoid repelled regions. (b) Analytical scalings (Table 2) confirmed for both small and large noise for 1-o (blue) and 1.5-o (orange) terms. The higher impact of the 1.5-o terms has been explained by its $\sim k_0^2$ as opposed to k_0 for 1-o, where $|k_0| \sim R_g/\sigma_d > 1$. The gap at $e_t = 1$ is due to sharp activation of noise back-steps during denoising for $e_t > 1$.

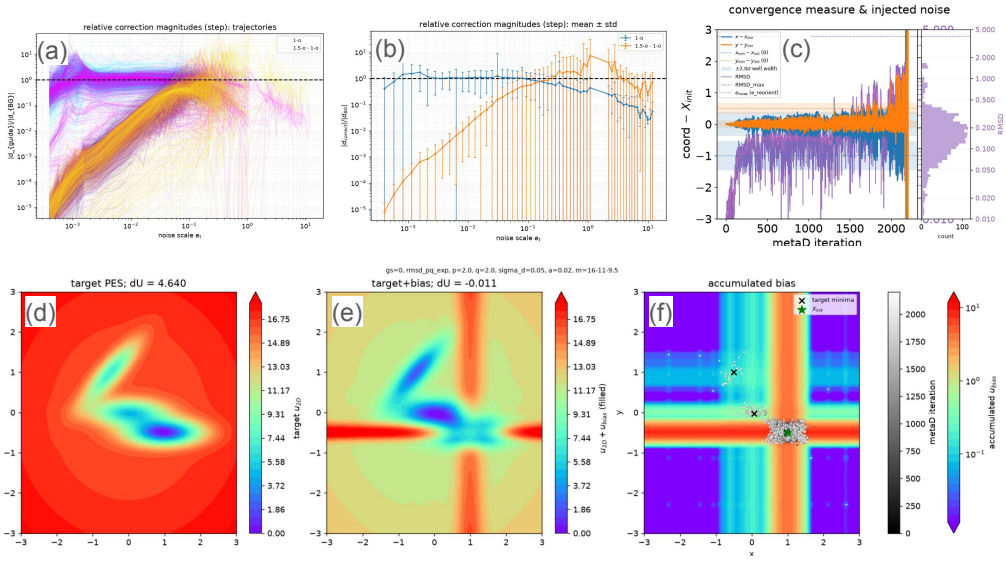

Figure 9: Demonstration of PBmetaD bias instead of localized gaussians. This is a known approach for high-dim systems. (top-left) No systematic strengthening of the guidance drive with bias accumulation, indicating the denoising trajectories are still avoiding repelled regions. (top-center) Both 1-o and 1.5-o terms obey the perturbative scalings, and are even weaker on average for intermediate noise. (top-right) $RMSD(t)$ and its distribution, showing the main peak for the original well and 2 separable peaks for 2 other basins.

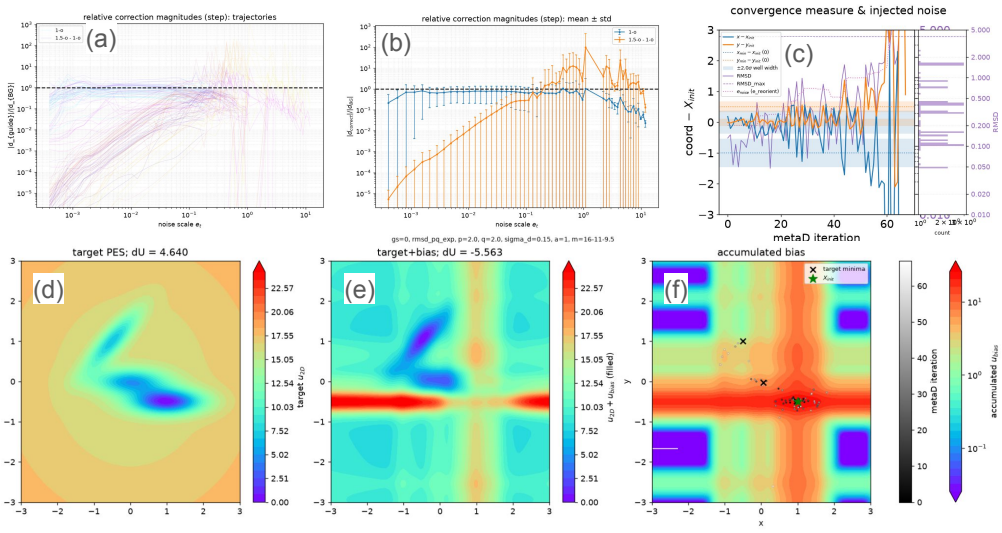

Figure 10: Using stronger individual bumps does not break the perturbative scalings (top-left + top-center) or the gradual nature of basin discovery (top-right + bottom-right). However, the overall time (number of metaD steps) to blow-up is significantly shorter, as expected since blow-up happens when all basins are filled-up, which happens in $N_{blowup} \sim dU_{basin}/a_{metaD}$ steps, or, more accurately, $N_{blowup} \sim (\sum_b dU_b W_b^2)/(a_{meatD}\sigma_d^2)$. Additionally, basin shadowing is observed from the per-DOF nature of the bias for such wide (large σ_d) bumps.

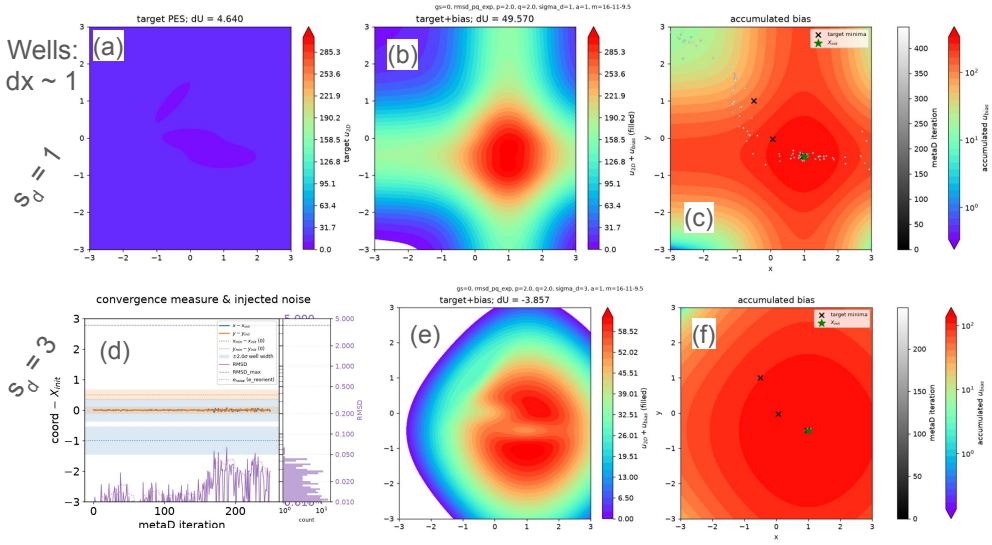

Figure 11: Too wider bumps disrupt discovery. (top) Bumps $\sigma_d \sim W_{well}$: approximately gradual discovery happens, but a lot of false-discovery is induced by artifacts from wide per-DOF biases. (bottom) Bumps $\sigma_d > W_{well}$: destabilization is not promoted since adding such wide biases is nearly identical to adding a constant, i.e. doing nothing.

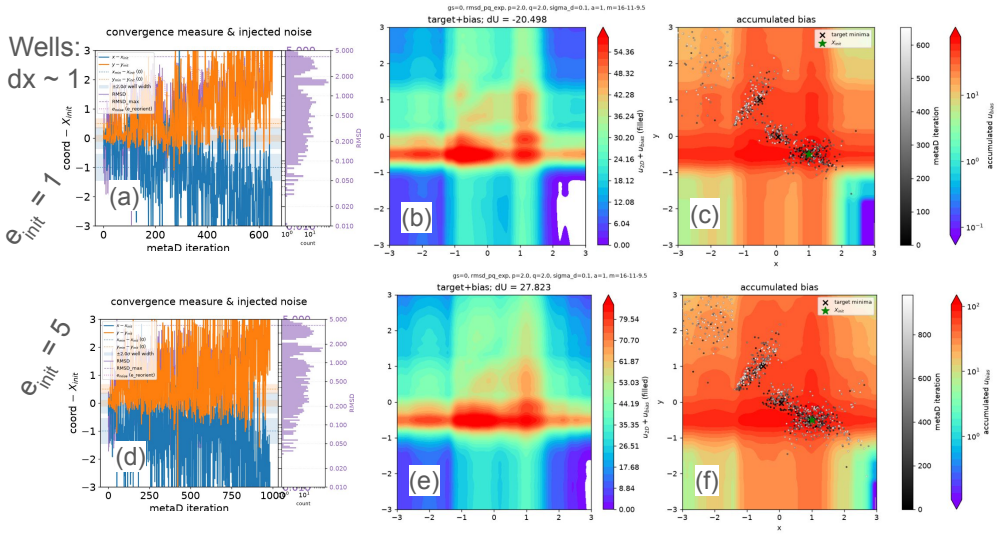

Figure 12: Taking $e_{init} \sim W_{inter-well}$ (top) and even more so $e_{init} > W_{inter-well}$ (bottom) allows to quickly discover all basins within that e_{init} but does not provide the information about the order in which they would be discovered.

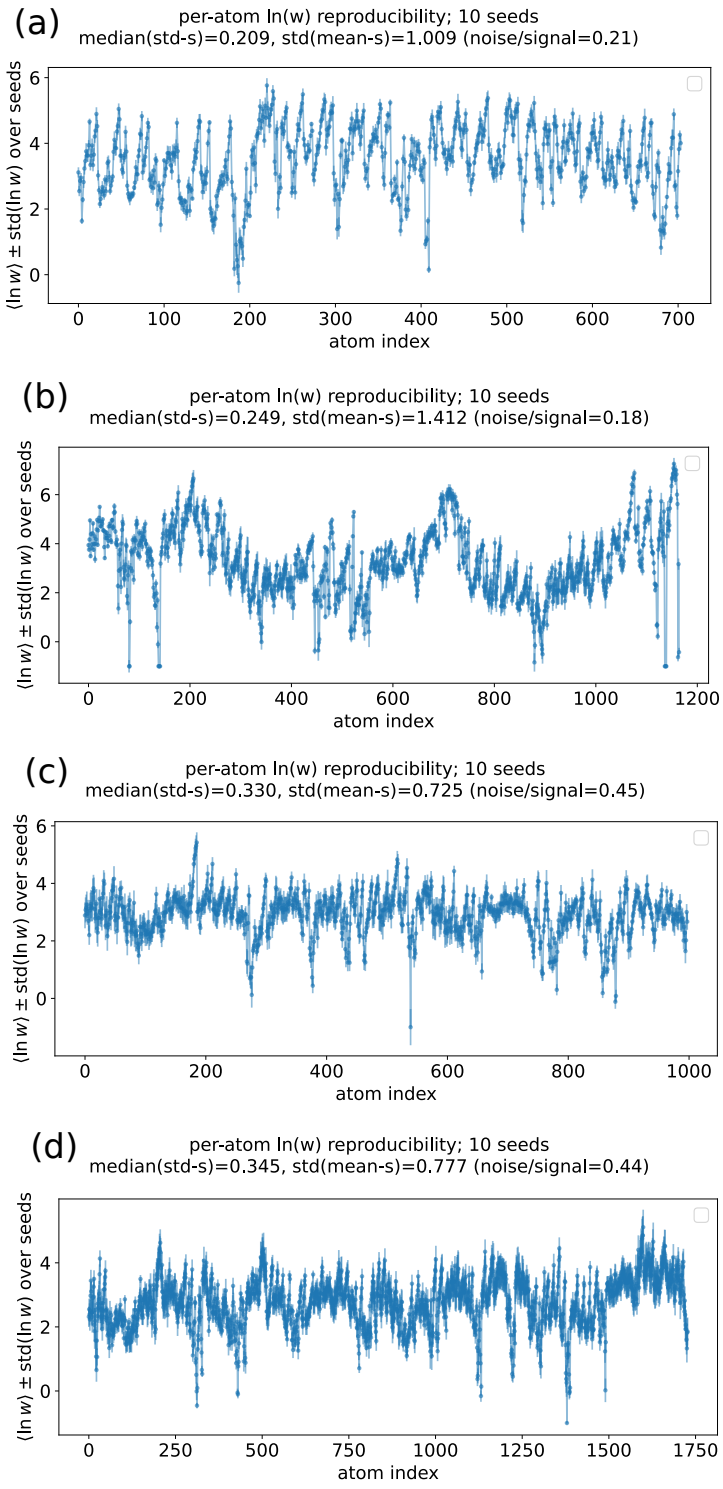

Figure 13: (a-d) are for mzb1-binder, rucaparib-binder, penguinpox-binder and pdl1-binder correspondingly. The variations of the PN-scores of each atom in the protein with its index and between 10 independent destabilization runs. Clear regions of atoms with high stability (e.g. near index $\sim 800 - 1000$ on (b)) and low-stability (e.g. $\sim 650 - 750$) emerge, implying the process is able to capture the structural features.

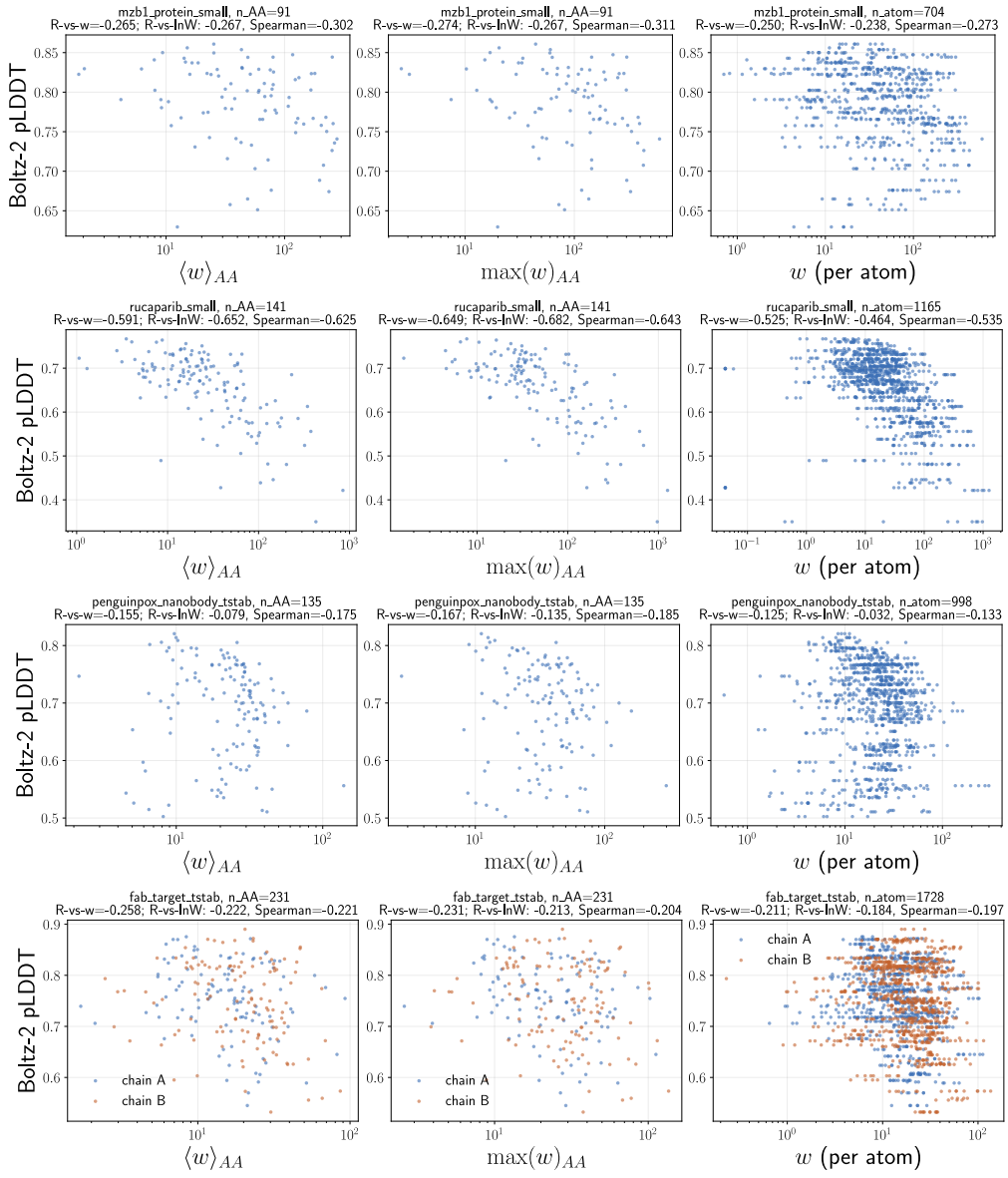

Figure 14: Complementarity of pLDDT (a simple metric) and PN-scores produced by RipGuide. Each row corresponds to a different system (top-to-bottom): mzb1-binder, rucaparib-binder, penguinpox-binder and pdl1-binder, respectively. Columns correspond to correlation of pLDDT with different PN-score-related metrics. PN-score is natively per-atom, but pLDDT from Boltz2 is per-AA, so: left is PN averages over each AA, middle is max PN for each AA, right is per-atom PN and pLDDT of each AA is broadcasted to all its atoms.

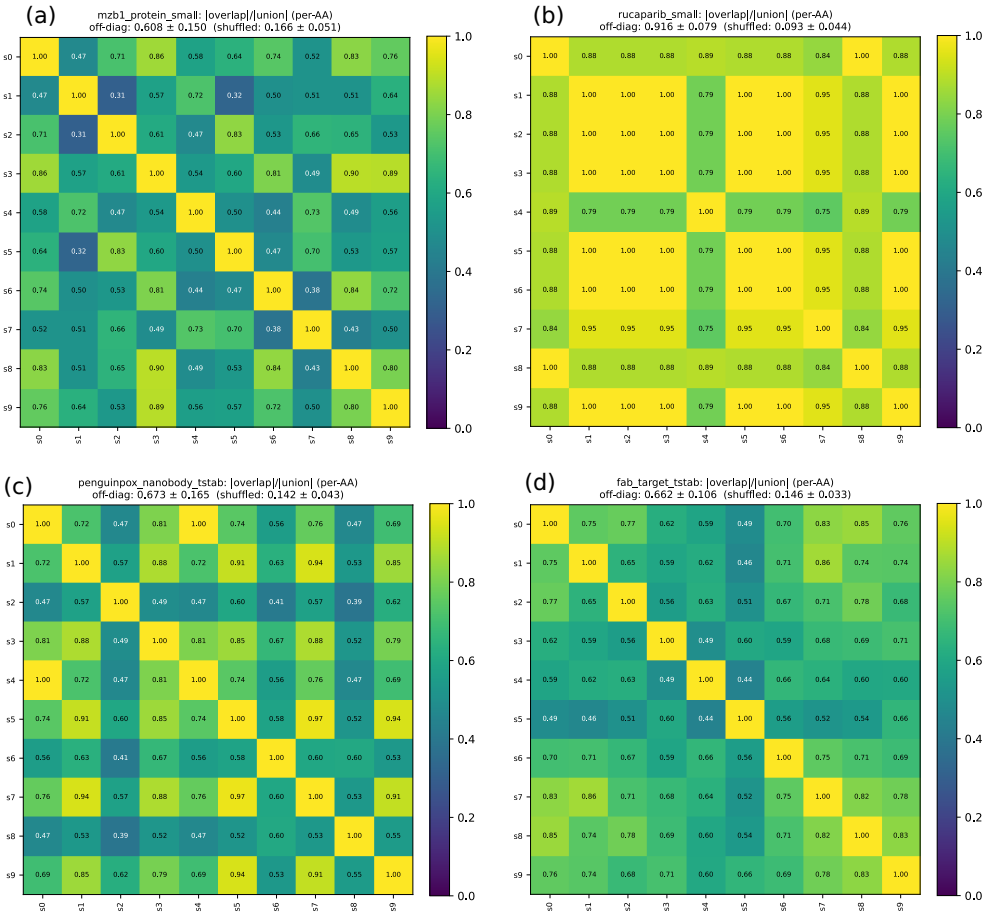

Figure 15: (a-d) are for mzb1-binder, rucaparib-binder, penguinpox-binder and pdl1-binder correspondingly. Given the per-atom PN-scores, DBSCAN clusters atoms with weights=PN-scores. Different RipGuide runs produce somewhat different weightings, which result in different AAs labeled at weak-link regions. But the clusters largely overlap, which is quantified in this figure, which shows |overlap|/|union| for every pair of weak-link AA sets. For all 4 cases, mean off-diagonal overlap are significantly higher than control means over shuffled AA indexing.