

SPICE: a simulator as teacher for learning beyond native-label coverage

Kuang Jingwen

redelectricity@outlook.com

Independent Researcher <https://orcid.org/0009-0008-4750-0674>

Article

Keywords: simulator-in-the-loop learning, molecular dynamics, reinforcement learning, protein stability, environmental conditioning, sequence intervention

Posted Date: September 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10978489/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

SPICE: a simulator as teacher for learning beyond native-label coverage

Author: Jingwen Kuang

Affiliation: Independent researcher

ORCID: 0009-0008-4750-0674

Code Organization: <https://github.com/SPICE-Protein>

Abstract

Machine learning for the physical sciences learns from corpora and inherits their coverage limits: a model trained on measured conditions may lack supervision outside the labelled regime. All-atom simulators can evaluate some such conditions but are often used only to score completed predictions or to generate static offline labels. We present SPICE as a simulator-supervised learning framework in which learner-generated proposals are evaluated online and physical feedback redirects subsequent search. The learner proposes state-condition interventions, the simulator returns graded physical observations and failure signals, and a critic translates these signals into action-value feedback for the actor. A task-specific admission rule determines which states are retained, while accepted trajectory conformations can be recycled as confidence-weighted pseudo-labels. The loop therefore extends supervision beyond direct training labels, subject to the simulator's domain and buildability constraints.

We demonstrate the framework in a label-sparse sequence-environment setting, where a compact structural prior seeds a hybrid evolutionary-strategy / soft actor-critic controller and an all-atom molecular-dynamics engine evaluates learner-generated proposals under configurable conditions. A prior-scale feasibility ablation using released configurations trained on 10 and approximately 45,000 chains produced 418 versus 338 admission records, corresponding to 338 versus 253 unique accepted sequences, showing that both priors can initiate the simulator-supervised search despite their large difference in prior scale. This is a configuration-level feasibility result rather than a corpus-size-only causal estimate. The full-corpus prior also retains contact-level signal on leakage-audited CASP14 targets (contact AUC 0.877), consistent with transferable contact information. The model, engine, logs, pseudo-labels and checkpoints are released. The verification notebook supports result-level audit of the reported arrays and summary statistics, whereas full stochastic RL/MD reruns require the released CPU environment and configurations.

The same learning regime can be instantiated in other sparse-label scientific systems when an evaluable simulator and a computable admission objective are available. This study establishes the loop in one concrete sequence-environment setting while defining its components independently of any single application domain.

Keywords: simulator-in-the-loop learning; molecular dynamics; reinforcement learning; protein stability; environmental conditioning; sequence intervention

1. Introduction

Many computational problems require prediction and intervention across continuous physical conditions, yet direct observations cover only a small and uneven subset. Data-driven models make inference efficient but inherit those coverage limits, whereas physical simulation can provide feedback beyond direct labels at substantial cost. SPICE uses a physical simulator as an online teacher rather than only as a post-processing filter, allowing a learned system to explore simulator-evaluable conditions and propose screened interventions.

Offline simulator labelling can provide additional training examples, but the resulting dataset is fixed before the learner encounters its own evolving proposal distribution. It may therefore spend simulator evaluations on states that are irrelevant to the learner's current search or fail to expose newly emerging failure modes in time to redirect that search. SPICE instead evaluates learner-generated proposals online. The simulator returns graded physical and chemical observables, rather than only a pass or fail label, and these signals are translated by the critic into feedback that the actor can use to adjust subsequent actions. The loop can therefore fail fast on unproductive branches while retaining directional information about the configured reward and admission objective.

Protein sequence–environment screening provides a useful test case because pH, temperature and ionic environment alter sequence behaviour, while structural databases rarely provide dense environmental labels. Molecular dynamics (MD) can evaluate a parameterized model where experimental labels are absent, but directly simulating a large sequence–environment space is expensive.

This setting motivates four linked questions:

- **Description.** How does a sequence respond to changes in its physical environment?
- **Physical validation.** Does the proposed sequence–environment state remain physically admissible over the configured simulation window when the environment is perturbed?
- **Intervention.** When a state encounters an operational collapse, can the system propose sequence changes for further physical evaluation?
- **Methodology.** Can a learning system connect directly to a physics engine that evaluates its own proposals online and uses the resulting evidence to redirect subsequent search?

The structural prior initializes exploration. Two optimization loops drive the search. An evolutionary-strategy (ES) macro-loop proposes discrete sequence and environment

changes, while a soft actor-critic (SAC) micro-loop applies continuous bias forces and environment offsets in response to physical feedback. Both are standard optimization methods adapted here to the simulator-in-the-loop setting. In the RL pathway, the simulator is the physical teacher, the critic acts as an intermediary that translates simulator observations into a learned action-value signal, and the actor uses that signal to update its subsequent actions. This division gives the learner directional feedback rather than a binary verdict alone.

Pre-training provides a minimally viable structural seed; the post-training simulator loop generates the additional environmental supervision evaluated here. This study demonstrates the teacher–learner–judge–recycle loop in a data-sparse protein regime.

2. Results

2.1 A compact prior provides an environment-conditioned structural initialization

The pre-training stage supplies the loop’s descriptive component: a sequence-and-environment model that represents a candidate conformation before physical interaction. Trained on roughly 45,000 PDB chains with a binned $C\alpha$ -distance objective, the prior reached an in-domain contact AUC of 0.866. On nine leakage-audited CASP14 targets it reached contact AUC 0.877 and $P@L/5 = 100\%$, indicating that the contact-level signal was not confined to the training examples.

We use the coordinate output as a coarse initialization, not as a competing native-fold predictor. The frame-based decoder enforces an approximately 3.8 Å virtual $C\alpha$ – $C\alpha$ bond. Distance-aware $SE(3)$ recycling and coordinate-distogram consistency make longer-chain outputs more usable as starting representations. Held-out coordinate quality remains coarse. This supports the division of labour tested here: the model provides a topological prior, and the simulator provides physical refinement and adjudication. The prior is a substrate for the loop, not its final scientific claim.

2.2 Path A probes conditions beyond direct corpus supervision

We evaluated sequence–environment conditions beyond the direct environmental labels available in the pre-training corpus. The configured engine was queried across its bounded pH 2–10 and 260–330 K domain at 1.0 bar. The reported runs sampled both ends and the interior of this domain. Operational failures were concentrated near the pH edges, notably pH 10.0, but also occurred in the interior; the record near pH 2.03 is a single exploratory point. The fixed-loop comparison in §2.5 shows both the $N = 10$ and $N = 45,000$ prior-scale configurations entering this same tested envelope.

The resulting records provide simulator-generated supervision at conditions without direct environmental labels. The physical meaning and implementation of this screening protocol are specified in Methods §4.3–4.4.

2.3 Path B generates environment-conditioned intervention hypotheses

Path B generated sequence-intervention hypotheses at the operational collapse conditions identified by Path A. In the first 7QF3 miniSOG intervention episode, five candidates completed the configured MD window and passed the structural admission criterion, with Q values of 0.90–0.93. The five candidates came from a single episode.

Frequently proposed positions included 50, 100, 102 and 103. Proposal chemistry varied across the tested environments. Near-neutral proposals included charge addition such as $V102 \rightarrow E$. At alkaline conditions near pH 10, accepted candidates included substitutions at the exposed N-terminal E2 position that removed or reversed its charge. At strongly acidic conditions near pH 2, later descendants were enriched for $E2 \rightarrow Q$, $K3 \rightarrow T$ and $L96 \rightarrow V$.

These patterns are descriptive outputs of the released search configuration and are reported as computational intervention hypotheses rather than experimentally validated rescue mutations.

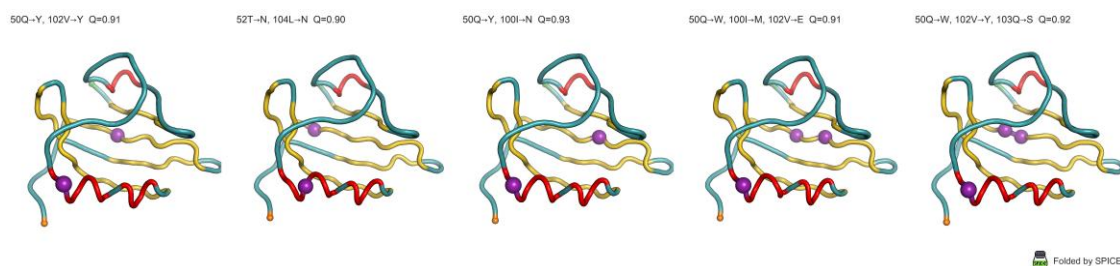


Figure 2. 7QF3 candidate structures from the first intervention episode of §2.3. The structures retain native-contact topology under the reported short-window criterion and represent computational intervention hypotheses, not experimentally validated rescue structures.

2.4 Simulator interaction generates and recycles supervision beyond native-label coverage

Accepted survivor conformations were recycled as confidence-weighted pseudo-labels in the released loop. For the $N = 45,000$ checkpoint, reflow changed the held-out CASP14 coordinate metric GDT-TS from 0.041 to 0.048 by the SMACOF-reconstruction route, while contact AUC remained near 0.88. In an accompanying verification run, the same checkpoint gave contact AUC 0.877 and $P@L/5$ 100% before reflow; an independent reflow archive recorded contact AUC 0.882 to 0.885 and $P@L/5$ 100 to 96.2. Without the anti-forgetting merge, reflow of the $N = 10$ seed changed survivor-fold reconstruction but reduced held-out contact AUC from 0.570 to 0.470.

Simulator-derived conformations were recycled in the released runs, and the coordinate metrics changed after reflow.

2.5 A prior-scale feasibility ablation initiates simulator-supervised exploration

We next performed a prior-scale feasibility ablation to ask whether simulator-supervised exploration can be initiated from a highly data-limited structural prior. We compared a prior trained on 10 chains with one trained on approximately 45,000 chains. This is an ablation of prior scale for loop initiation under two released end-to-end configurations. The ablation tests whether both prior scales can initiate the loop and how the resulting proposal and admission records differ.

Both priors were evaluated at conditions spanning the tested pH 2–10 and 260–330 K bounded envelope. Across 7QF3, lysozyme structures 6QQE/8D8F and RNase A 1JVT, the $N = 10$ configuration produced 418 accepted admission records with median $Q = 0.849$, compared with 338 records and median $Q = 0.819$ for the $N = 45,000$ configuration. The $N = 45,000$ prior enriched tryptophan substitutions in some proteins, whereas the data-limited prior sampled a broader mixture, including paired-cysteine and proline substitutions.

The $N = 10$ run produced 418 records and 338 unique accepted sequences, with 80 repeated sequence admissions. The $N = 45,000$ run produced 338 records and 253 unique accepted sequences, with 85 repeated sequence admissions. Collapsing each run to unique sequences leaves the direction unchanged, with median Q values of 0.851 and 0.824, respectively. The corresponding record-level and unique-sequence-level rank summaries are reported in Supplementary Table S2.

Both candidate sets were evaluated by the same simulator and Q -gate. Within their respective released configurations, the $N = 10$ prior entered the bounded simulator screen despite its minimal native-label seed, although proposal chemistry and admission-record counts differed (Fig. 1).

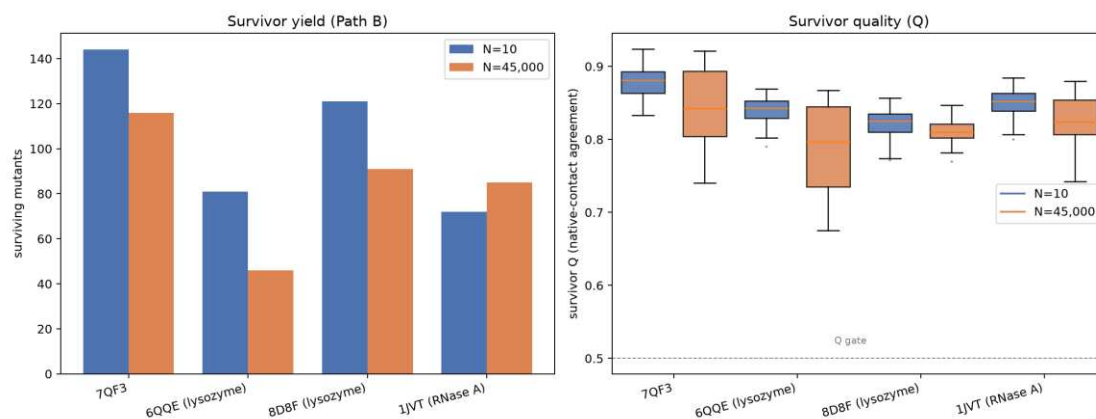


Figure 1. Prior-scale feasibility ablation. The $N = 10$ and $N = 45,000$ priors were evaluated across the bounded tested environment envelope under their released configurations. Admission records include repeated sequence admissions.

3. Related work and methodological positioning

3.1 Data-driven structure and stability prediction

AlphaFold2, AlphaFold3 and ESMFold substantially advanced sequence-based structure prediction, particularly for native-like or training-distribution conditions [1–3]. Sequence-based mutation predictors and language-model approaches offer useful estimates from learned sequence statistics [4]. These systems address questions complementary to SPICE: it does not seek to replace high-accuracy native-condition structure prediction, but to use a physical simulator for environment-conditioned stability screening and intervention where direct labels are sparse.

3.2 Environment-conditioned protein modelling

Temperature-aware modelling and mutation-design methods occupy part of the environment-conditioned space [6–9], while constant-pH molecular dynamics offers a physics-based route for pH-dependent simulation [10]. To our knowledge, the combination of continuous environmental conditioning, online all-atom-MD feedback and mutation intervention in a closed learning loop remains insufficiently explored. Our claim concerns the tested combination and implementation, not the absence of related components in prior work.

3.3 Reinforcement learning for protein design

Reinforcement learning has been used for adaptive sampling, therapeutic design and online protein generation [11,12,16]. These systems demonstrate the value of sequential decision-making, but generally rely on learned, statistical or coarse rewards rather than on an all-atom MD simulator acting as both online teacher and candidate adjudicator. SPICE-SAC addresses the practical consequences of slow feedback, sparse collapse events and mixed continuous–discrete actions.

3.4 Molecular dynamics as a design or fitness function

Rosetta established the value of physics-grounded computational platforms using authored scoring and search procedures [5]. Other automated design pipelines have used MD-derived fitness or physical relaxation, including one-shot generate–score–optimize workflows [19]. SPICE differs in the closure of its loop: a learned surrogate proposes actions, online MD supplies physical observations and failure signals, and conformations accepted by the controller are written back as pseudo-labels. The novelty lies in the closed loop, not in using MD as a score by itself.

3.5 Positioning of the present study

Existing approaches address different parts of the four-step methodological chain: structure predictors describe a state, stability predictors estimate selected effects from

data, MD-based workflows evaluate physical states, and reinforcement-learning systems search an action space. SPICE connects these roles in one simulator-guided learning and intervention framework instantiated in a protein sequence–environment screening problem. It does not replace AlphaFold-class coordinate prediction, Rosetta-style design, experimental screening or long-timescale molecular simulation. Its contribution is the demonstrated teacher–learner–judge–recycle workflow and its analysis of how physical screening, prior scale, auditability and algorithmic search interact in the tested protein systems.

4. Methods

4.1 Problem formulation and objective

The loop is defined over an instantiation comprising a perturbable state, steerable conditions and a simulator that returns the physical consequence of each state–condition pair. We demonstrate the system using proteins as a concrete example, where the state is an amino-acid sequence, the conditions are environmental scalars such as pH, temperature and ionic strength, and the simulator is an all-atom molecular-dynamics engine. We state the formalism for this protein instantiation while keeping the loop definition independent of protein-specific variables. The same four-stage structure may provide a starting point for future work on other sparse-label systems with computable physical objectives.

Let s denote an amino-acid sequence, $e = (\text{pH}, T, I)$ an environmental state and $\mathcal{S}$ the parameterized molecular-dynamics simulator. At time t , the learner proposes an action a_t , the simulator advances the physical state and returns an observation

$$y_t = \mathcal{S}(s, e, a_{0:t}) = \{U_t, \mathbf{M}_t, Q_t, T_{\text{kin},t}, \dots\}, \quad (1)$$

where U_t is the instantaneous potential energy, $\mathbf{M}_t$ is a vector of physical indicators and Q_t is the native-contact fraction when a reference structure is available. The action space contains continuous bias-force coefficients and environment offsets, and, at scheduled intervention steps, a discrete mutation position and target amino acid.

The v1 objective is not long-timescale free-energy estimation. It is to identify sequence–environment pairs that remain physically admissible over the configured online window while retaining the parent or wild-type contact topology. For a fixed sequence, Path A searches the environment space. If it identifies a collapse condition e_{fail} , Path B searches mutated sequences s' at that condition. For candidate i , the ES fitness is

$$F_i = N_{\text{steps},i} Q_i, \quad (2)$$

where $N_{\text{steps},i}$ is the number of MD steps completed and $Q_i = Q(s'_i, s_{\text{ref}})$ is its native-contact agreement. A candidate is accepted only if it completes the configured window

and $Q_i \geq q_{\text{gate}}$, with $q_{\text{gate}} = 0.5$ in the reported runs. Here Q is a structural proxy, not a direct measurement of biochemical function.

4.2 Overall teacher–learner–judge–recycle workflow

The complete v1 workflow is summarized in Fig. 3. The diagram separates the data-derived prior, the learned controller, the Rust physical teacher and the two sequential intervention paths. Unlike offline simulator labelling, in which simulated examples are generated before learning and then treated as a fixed training resource, the SPICE simulator evaluates learner-generated actions during the search. Its outputs enter the reward, replay, admission and recycling pathways before later proposals are generated, coupling supervision to the learner's evolving proposal distribution.

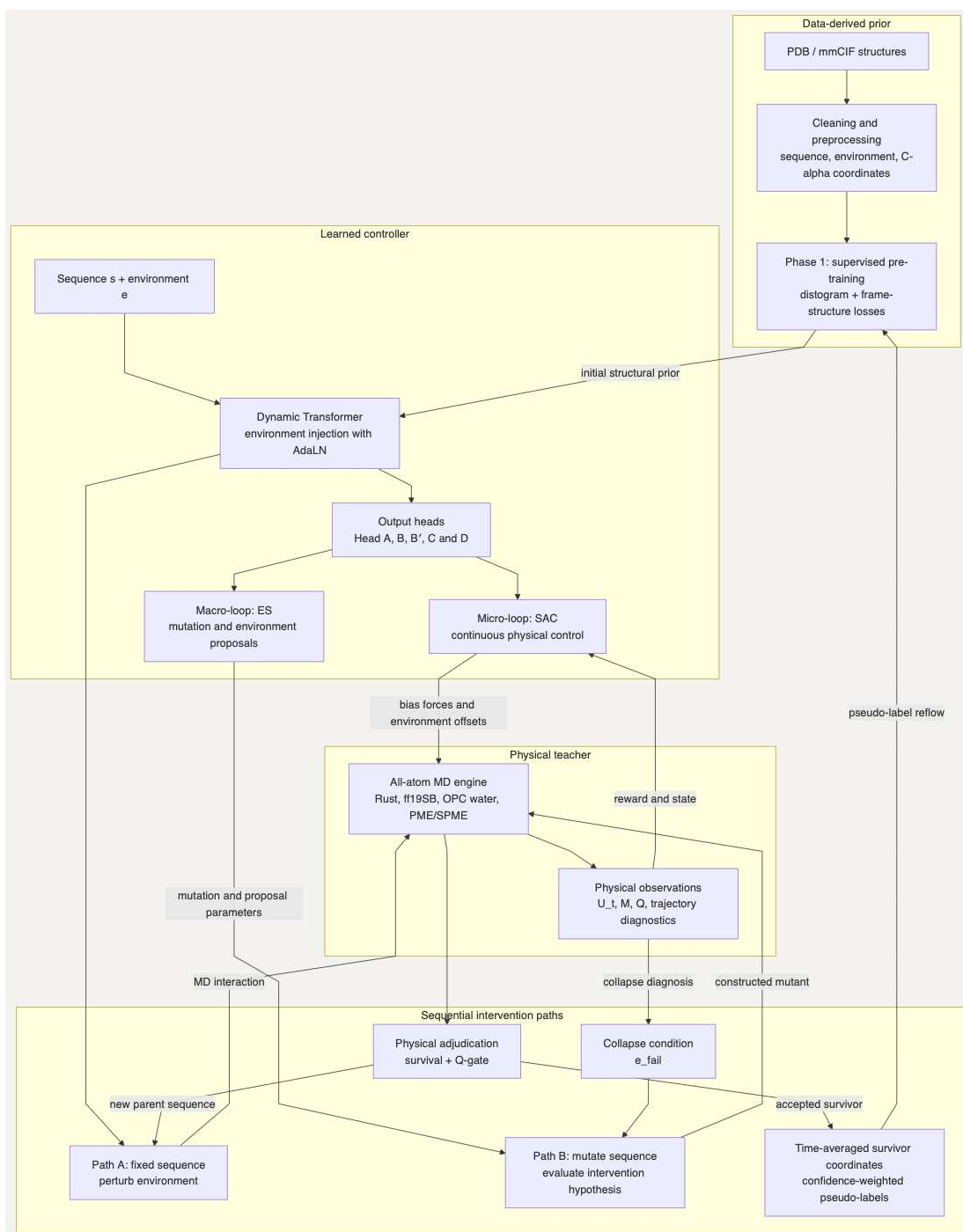


Figure 3. The SPICE teacher–learner–judge–recycle loop. The sequence model supplies a structural prior and proposals; Path A probes environmental failure; Path B searches for sequence interventions; the all-atom MD engine supplies online physical feedback and candidate adjudication; accepted conformations return as pseudo-labels and new parent sequences. The protein system is the current validation instantiation.

SPICE uses a four-stage closed loop.

1. Prior initialization. A dynamic Transformer is pre-trained on cleaned protein structures from PDB that have environment tags. The principal target is a binned $C\alpha$ -distance distribution; a frame-based coordinate head supplies a coarse structural initialization.
2. Physical screening and Path A. A parent sequence is built in the MD engine and screened for physical legality. Path A then evaluates separately built, fixed-protonation environmental parameterizations while perturbing pH and temperature in both directions around the current state, and the SAC controller applies continuous actions. The run records surviving trajectories and operational collapse conditions.
3. Path B intervention. When Path A reaches a collapse condition, the ES macro-loop samples one-to-three-residue mutations from Head B. Each candidate is constructed, equilibrated and evaluated at the same e_{fail} , using the frozen Path-A SAC actor for the within-trajectory control problem.
4. Physical adjudication and recycle. Candidates that complete the MD window and pass the Q-gate are retained. Their time-averaged coordinates are added as confidence-weighted pseudo-labels, and accepted descendants can become parents for the next generation. Path-A failures are retained as negative environmental outcomes rather than treated as successful training examples.

The simulator plays four distinct roles: it supplies the online teacher signal, returns physical observations and failure signals for candidate interventions, generates structural examples outside the original corpus, and provides observable feedback that support artifact-level auditing of a recommendation. The critic translates these observations into action-value feedback for the actor, while the controller applies the configured survival and structural-retention admission rule and processes accepted trajectory conformations into pseudo-labels. The defining feature is that these roles close into one loop, rather than MD being used only as post-processing.

4.2.1 Design evolution and scope of comparisons

SPICE v1 was developed through sequential failure-driven design. The initial mutation-only prototype, represented by the first mutation exploratory run, exposed the need to make structural retention explicit; the Q-gate and conservation mask were then introduced as structural-retention constraints. Subsequent numerical failures near difficult environmental conditions motivated explicit crash handling, clipping and non-finite-value safeguards. The current released configuration is the endpoint of this engineering sequence, while the $N = 10$ versus $N = 45,000$ experiment provides a separate prior-scale feasibility ablation for loop initiation.

4.3 Protein-system instantiation and data pipeline

The current instantiation uses a variable-length amino-acid sequence and an environment vector containing pH, temperature and ionic strength. pH and temperature are actively perturbed in the reported experiments; ionic strength is represented in the model state but is not systematically explored as an independent active axis. The principal post-training systems are 7QF3 miniSOG, the two lysozyme structures 6QQE and 8D8F, and bovine RNase A structure 1JVT. The $N = 45,000$ post-training configuration accepts sequences of 70–150 residues, whereas the $N = 10$ configuration uses 80–150 residues. The reported systems were selected from the overlapping 80–150-residue range.

The pre-training data pipeline downloads PDB records, cleans chain and residue order, extracts sequence and $C\alpha$ coordinates and writes the resulting (s, e, x) records through parquet and TFRecord intermediates. The default corpus contains approximately 45,000 chains. Where experimental environmental annotations are absent, default native-like values are used for the pre-training environment input; an annotated `has_env` subset can be selected separately.

4.4 Molecular-dynamics teacher

4.4.1 Engine and interface

The v1 engine is a modified fork of the open-source Rust dynamics repository [17]. It retains the Amber ff19SB force field, OPC water and PME/SPME electrostatics [13–15], and adds physical indicators, bounded-domain scanning and hooks for RL bias forces and state transfer. For each requested pH, v1 builds a system with a fixed discrete protonation assignment rather than dynamically titrating during a trajectory. The assignment uses approximate intrinsic pK_a values (Asp 3.9, Glu 4.2, His 6.0, Cys 8.3, Lys 10.5 and Tyr 10.5); arginine and the termini are not titrated. The pH parameterization is therefore not constant-pH or replica-exchange MD. The v1 histidine path has an additional implementation limitation: geometry-based tautomer resolution can override pH preselection and returns only HID or HIE, so the acidic HIP state is not guaranteed in the effective force-field assignment. Hydrogen placement and residue-variant typing can consequently be inconsistent in affected builds, and the hydrogen set can be misassigned, leaving some histidine-containing proteins unbuildable, as described for 7RSA in Supplementary Methods S6. We do not claim authorship of the underlying force field or base integrator. The engine is used here as an implementation-dependent physical-legality label generator rather than as a general-purpose molecular-dynamics package.

4.4.2 Physical observables

At each observable checkpoint the engine returns the potential energy and

$$\mathbf{M}(t) = [m_1(t), m_2(t), m_3(t), m_4(t), m_5(t)]^T. \quad (3)$$

The components are:

Component	Definition	Interpretation
m_1	$\frac{\text{MAD}(U)}{k_B T}$, with the MAD scaled by 1.4826	Robust potential-energy fluctuation
m_2	$\frac{ R_g(t) - R_g^{\text{ref}} }{R_g^{\text{ref}}}$	Radius-of-gyration drift
m_3	$1 - \frac{SS_{\text{kept}}(t)}{SS_{\text{ref}}}$	Secondary-structure loss from DSSP-lite main-chain hydrogen bonds
m_4	Heavy-atom pair fraction with $\frac{d_{ij}}{r_i + r_j} < 0.6$	Atomic clash fraction
m_5	Mean surface absolute difference between the current charge of surface ionizable residues and their neutral-pH reference charge	Surface-charge mismatch

The engine also exposes kinetic temperature, acceleration-clamp and observable per-residue force for diagnostics and optional strain penalties. Surface-residue identification for m_5 uses the documented burial-count approximation. In the v1 engine, the reference charge for m_5 is fixed at neutral pH 7.0 rather than evaluated at the current pH of the simulation, so m_5 measures deviation from a neutral-pH ideal state and is not itself a pH-dependent observable; this should be kept in mind when interpreting the m_5 correlation reported in Supplementary Note S6.

4.4.3 Reward, crash and survival criterion

The default step reward is

$$R_{\text{step}} = -\frac{U_t}{U_{\text{ref}}}, \quad U_{\text{ref}} = 10^5 \text{ kJ mol}^{-1}, \quad (4)$$

clipped to $[-10, 10]$, with a terminal crash penalty of -100 . A crash is an energy that is non-finite or that exceeds a magnitude threshold; the Rust engine itself terminates at a potential energy above $10^8 \text{ kcal mol}^{-1}$, while the Python reinforcement-learning wrapper used for the reported runs treats a non-finite energy or $|U_t| > 10^7 \text{ kJ mol}^{-1}$ (about $2.4 \times 10^6 \text{ kcal mol}^{-1}$) as a crash, so the effective in-run threshold is the Python-side one. A trajectory that reaches the episode maximum without a crash receives a terminal reward of zero. The library default episode maximum is 200 steps. The reported data-efficiency deployments use 10 equilibration steps (20 fs) followed by 20 measured steps at a 2 fs time step (40 fs of measured dynamics, 60 fs elapsed).

A survivor is a candidate that completes the configured short window without catastrophic energy failure and passes the Q-gate. This is an admission label under the v1 protocol. The criterion detects physical legality and early strain, including severe clashes or electrostatic explosions.

4.5 Sequence model

4.5.1 Dynamic Transformer and environment conditioning

The v1 learner is a dynamic Transformer encoder with padding and attention masks, sinusoidal positional encodings and adaptive layer normalization. The normalized environment vector is injected into every Transformer layer through learned scale and shift parameters. The shared sequence–environment representation is denoted by z . In v1, Head B' reuses the Head-A frame decoder for the mutated sequence and has no independently trained coordinate parameters.

4.5.2 Output heads

The model contains the following heads:

Head	Output	Role
Head A	$C\alpha$ coordinates	Fixed-sequence structural initialization for Path A
Head B	Position-by-amino-acid mutation probabilities	Mutation proposal for Path B and ES (Masked-ES below)
Head B'	Mutant $C\alpha$ coordinates	Head-A frame-decoder output for the mutant sequence; no independent v1 weights
Head C	$[\Delta pH, \Delta T]$	Active environment perturbation
Head D	Two sigmoid confidence values	Path-A and Path-B confidence estimates
Distogram head	Binned pairwise $C\alpha$ distances	Principal pre-training target

Head A uses a frame-based decoder: per-residue orientations and local directions are integrated along the chain, enforcing an approximately 3.8 Å virtual $C\alpha$ – $C\alpha$ bond by construction. A distance-aware $SE(3)$ recycling module uses predicted distance information to improve the workability of the coarse coordinate prior. It is an initialization for the physical loop, not a claim of AlphaFold-level structure prediction.

4.5.3 Pre-training objective and length gating

The principal pre-training loss is a symmetrized binned-distogram cross-entropy over valid residue pairs, using 48 bins spanning approximately 3 – 48 Å. Coordinate supervision combines Kabsch-aligned $C\alpha$ error, a chirality term and a non-adjacent $C\alpha$ clash penalty. Coordinate, chirality and clash losses are applied only to chains of at most 200 residues (coord_max_len=200), because cumulative frame integration causes orientation errors to grow with length and can otherwise dominate the shared encoder. The distogram objective is applied to longer chains as well.

The coordinate–distogram consistency loss and two-step distance-aware $SE(3)$ recycling provide a long-chain coarse prior without requiring the coordinate head to match a clean native fold.

4.5.4 Pseudo-label reflow

For an accepted mutant, the engine trajectory is time-averaged after equilibration. The resulting record $(s', e_{\text{fail}}, x_{\text{MD}})$ is added to the training stream with a confidence weight proportional to the observed survival fraction. In the reported reflow, pseudo-labels are merged with the original pre-training data to reduce catastrophic forgetting. The reflowed coordinate head is evaluated separately from the original prior so that improvement in the learned representation is not confused with the initial coordinate quality.

4.6 Hybrid ES–SAC controller

4.6.1 Path A and Path B scheduling

Path A is the fixed-sequence controller. It perturbs the environment in positive and negative directions, and applies SAC continuous actions while the Rust engine supplies feedback. In v1 these two directions are executed sequentially rather than concurrently. A collapse in either direction records e_{fail} and activates Path B.

Path B is the sequence-intervention controller. ES samples mutation candidates from Head B and evaluates them at e_{fail} . The Path-A SAC actor is reused and frozen during this candidate evaluation, so ES searches the sequence-level proposal mechanism while SAC handles within-trajectory continuous control. Accepted descendants can become parents for subsequent episodes.

4.6.2 ES macro-loop

The v1 macro-loop applies evolutionary search to the Head B, Head C and policy-selection weights. ES ranks, crosses and perturbs candidates without backpropagating through the MD engine, while the Transformer backbone remains frozen. SAC learns the local response to physical feedback.

4.6.3 SAC state and action spaces

The SAC actor receives $z + e$ and does not receive the physical-indicator vector. Its continuous head outputs $M = 16$ low-rank force-primitive coefficients together with $[\Delta pH, \Delta T]$, sampled from Gaussian distributions with reparameterization. The coefficients are mapped through a column-normalized primitive matrix $W \in \mathbb{R}^{[L,3] \times M}$ to per-atom bias forces, whose magnitude is bounded by a tanh transform to $[-0.5, 0.5] \text{ kcal mol}^{-1} \text{ \AA}^{-1}$.

The discrete head represents a mutation position and target amino-acid identity. It uses categorical distributions and a differentiable Gumbel–Softmax approximation during learning. These actions are concatenated for value estimation, but they do not have the same temporal frequency in the environment.

4.6.4 Privileged critic and temporal action masks

The twin critic receives privileged physical information:

```
critic input = z + environment + M + u_hist + continuous action + discrete action + mutation_mask  
actor input  = z + environment
```

Here u_hist is a ten-step potential-energy history window. The critic estimates future accumulated $-U_t$ and uses the smaller of its two Q estimates to reduce overestimation. The simulator supplies the physical evidence, while the critic translates that evidence into an action-value signal that the actor can use to adjust subsequent actions. The actor is deliberately prevented from directly observing $\mathbf{M}$, so that it must learn decisions from the sequence representation and environment rather than simply copying instantaneous physical diagnostics.

Mutation actions are applied every 20 steps, whereas bias-force and environment-offset actions can be applied at every step. The environment masks the discrete head when `mutation_allowed = (step % 20 == 0)` is false. An engine-side action mask rotates the residues receiving bias forces at the mutation interval. The discrete entropy contribution is correspondingly gated in the critic target.

4.6.5 Replay, updates and numerical safeguards

Experience is written to a shared replay buffer and updates are interleaved with MD collection. The reported n45000 deployment uses batch size 32 and triggers an update after `update_every_steps=20`; the n10 comparison run uses batch size 256 and `update_every_steps=200`. Both runs use 20-step episodes, and their logs record `max_episodes=40` for the comparison. The implementation also clips extreme indicators and rewards, washes non-finite probabilities and Bellman targets, enforces mutation-mask shapes, bounds critic and entropy terms, guards local minimization against non-finite inputs and records repeated explosion states before engine construction. These safeguards protect numerical stability and exploration efficiency without changing the definition of the physical teacher.

4.7 Mutant construction and structural-retention constraints

4.7.1 Side-chain rebuilding and equilibration

Each candidate mutation is built by fitting backbone atoms, selecting a clash-minimizing target-residue rotamer and performing short NVT equilibration before reward measurement. Head B' reuses Head A for the mutant sequence, but its coarse coordinates are screened for finite values, radius-of-gyration anomalies and local $C\alpha$ clashes. In the reported v1 workflow, unbuildable predictions are rescaled when possible; otherwise, construction falls back to the parent or wild-type $C\alpha$ backbone with full side-chain rebuilding. If engine construction still fails, the wild-type-backbone pathway is retried. Thus, reported candidates are evaluated after side-chain rebuilding and MD equilibration.

When the predicted mutant backbone is unbuildable, the production path uses the parent or wild-type $C\alpha$ backbone fallback rather than assuming that Head B' coordinates are directly engine-buildable.

4.7.2 Q-gate and conservation mask

Native contacts are defined from a reference $C\alpha$ structure using an 8 Å distance criterion. Q is the fraction of reference contacts retained by the time-averaged mutant structure. In the reported runs the reference is the wild-type native $C\alpha$ structure of the starting protein (not the parent sequence of a given episode, which in later generations may itself be an earlier accepted mutant), so Q measures retention of the original fold. Path B ranks candidates using survival multiplied by Q , and rejects candidates with $Q < 0.5$ in the reported runs. A residue-type conservation prior, implemented as a fixed per-amino-acid table with a protection threshold of 0.8, protects strongly conserved positions from mutation. The reported v1 runs do not use externally supplied MSA conservation scores. The Q-gate and conservation mask are structural proxies for the configured admission objective.

4.8 Evaluation protocols

4.8.1 Folding-prior evaluation

The pre-training prior is evaluated using contact AUC, $P@L/5$, pairwise-distance error and coordinate metrics on in-domain data and leakage-audited CASP14 targets. The CASP audit excludes targets with direct structural entries in the pre-training corpus. Coordinate metrics are interpreted as initialization diagnostics, not as the main claim of the study.

4.8.2 Post-training evaluation

The post-training loop is evaluated using Path-A coverage and collapse events, Path-B acceptance and Q distributions, parent-offspring records, engine observable feedback and pseudo-label reflow. The $N = 10$ and $N = 45,000$ comparison is a prior-scale feasibility ablation for loop initiation. It uses the same Path-A/Path-B design, construction procedure, Q-gate and nominal environment window, while retaining the run-specific optimization settings in each released configuration. Pre-training supplies the initial structural seed; post-training generates additional simulator-derived supervision. The comparison is not a one-variable corpus-size causal ablation, and its single-run records are not estimates of run-to-run variance.

4.8.3 Published-data anchoring and short-window sanity check

For the reported published-data anchor, the engine builds solvent systems from public structures, performs local minimization for 50 iterations and evaluates pH 2–10 at fixed 298 K with three short repeated trajectories per condition. Each condition starts from a clean clone of the reference state and is classified as non-crashing when at least two of three repeats complete. This is a simulator-defined short-window buildability and legality check, not an experimentally calibrated stability boundary. No T_m or pH-boundary ranking

is claimed. A FireProtDB pilot probes whether the short-window signals behave like a calibrated mutation $\Delta\Delta G$ predictor. The null results for survival margin and potential-energy change are treated as a resolution boundary, while the m_5 signal is reported separately as an uncorrected exploratory pooled association. The trend-based early termination used in a longer-trajectory protocol is inert over the short windows used in the reported runs.

4.8.4 Statistics and reproducibility

Survivor native-contact agreement Q was compared between the two prior scales with a pre-specified two-sided Mann–Whitney U procedure, reported per protein and pooled in §2.5 and Supplementary Table S2. Because the admission records include repeated sequence admissions and are not independent replicates, the comparison is repeated after collapsing each run to unique accepted sequences (the maximum Q per sequence), and both the record-level and the unique-sequence-level results are reported. No run-to-run variance estimate is possible from the single released run per scale, so the statistics describe the two released configurations rather than a population of runs. Correlations between physical ranking signals and Q are reported by protein and prior scale because pooled values can be confounded by between-protein baselines. Released candidate and failure records, survivor coordinates, mutation-frequency tables, configuration files, leakage-audit scripts and job logs are provided with the data and described in Supplementary Methods.

5. Discussion

5.1 What this study establishes

SPICE provides an end-to-end protein-system demonstration of a simulator-supervised learning loop. The model supplies a structural prior and proposals; Path A identifies operational environmental failure conditions; Path B evaluates sequence interventions; and accepted MD conformations return as pseudo-labels. The results establish simulator-defined exploration beyond direct native-condition labels within the configured domain, not long-timescale thermodynamic stability, biochemical function, causal benefit over alternative loops or cross-domain transfer.

5.2 What the learner does and does not learn

The learner proposes and adapts, whereas the simulator supplies physical and chemical observations that enter the controller's reward and admission pathways. In the RL pathway, the critic translates these observations into action-value feedback that the actor can use to adjust subsequent actions. ES provides sequence-level proposal generation, while SAC provides within-trajectory continuous control. MD-averaged survivor conformations are then recycled to expose the coordinate head to structures formed under the explored environmental conditions. The Q -gate and conservation mask constrain structural

retention, so the objective is to produce physically admissible, fold-retaining outputs rather than de novo sequences whose physical failures are deferred to downstream wet-lab screening.

5.3 Data coverage versus physical interaction

The central distinction is between direct native-label coverage and simulator-supervised learning. Direct-label coverage records observations in the corpus, simulator-evaluable coverage records conditions the configured v1 engine can build and run, and neither is synonymous with experimentally validated stability coverage. The structural corpus supplies an efficient prior, whereas simulator interaction generates reward, admission outcomes and pseudo-labels at selected non-native conditions. The prior-scale feasibility ablation shows that both released configurations can enter the bounded simulator screen, while contact generalization and coordinate quality remain data-dependent. Physics-derived supervision therefore complements, rather than replaces, data-derived supervision, and its interpretation depends on force-field fidelity, numerical validity and the admission observable feedback.

5.4 When does the loop generalize?

The teacher-learner-judge-recycle loop provides a computational pattern that can be instantiated in other sparse-label scientific systems when the following conditions hold:

- **Evaluable simulation.** A physical simulator must be evaluable at conditions of interest cheaply enough to close a learning loop; for SPICE, this is the short-window all-atom screen.
- **Computable admission objective.** A criterion must separate an acceptable from an unacceptable physical outcome; here, this is survival and fold retention under the Q-gate. Where the objective is not reducible to computable observables, the pattern may not apply.
- **Label sparsity.** The target regime should be sufficiently label-sparse for interactive simulation to add value; otherwise, the data-derived prior may be the cheaper and better teacher.

When these conditions hold, the division of labour is the same as here: the learned model proposes, the simulator evaluates, and accepted outputs can be written back as supervision. What does not transfer automatically is fidelity—the simulator must faithfully model the governing physics for the new class—or relevance—the admission criterion must match what the user cares about. These are the same load-bearing assumptions that bound the protein demonstration, and stating them is what separates a reusable method statement from a single-system report.

5.5 Auditable computational hypotheses

A SPICE recommendation is an auditable computational hypothesis, not an opaque score or proven mechanism. The released record links each candidate to its survival status, Q outcome, physical observable feedback, sequence change, environment and construction pathway. This evidence trail makes charge-balance or contact-retention explanations testable by longer simulations, improved force fields or experiment without implying guaranteed causality.

5.6 Prior proposal versus physical admission

The prior proposes, whereas the simulator and Q-gate evaluate admission under the reported criterion. The internal control shows that prior scale changes initialization and mutation chemistry within the fixed search protocol. The resulting trajectories record algorithmic search rather than biological evolution. They show how simulator-defined admission shapes the reported search, without assigning biological meaning to the trajectory.

5.7 Open implementation supports an auditable physical signal

SPICE uses an openly released, all-atom simulator based on the Amber ff19SB force field and OPC water model [13,14]. The `spice_engine`, modified dynamics dependency, model code, preprocessing pipeline, configurations, tests, logs and derived artifacts are available for inspection and re-analysis. This open record supports artifact-level audit of the proposal distribution, admission criterion, pseudo-label recycle and failure modes, while the physical outputs remain conditional on the documented model and protocol.

5.8 Limitations and future work

The v1 study provides a resource-aware protein-system demonstration of the framework. Its short-window physical signal, build-time protonation model and released demonstrations define the scope of the present evidence. Detailed protocol limitations, external checks and configuration-level statistics are provided in Supplementary Methods. SEv1.5 is being developed as a separate architectural successor and is outside the scope of this study.

The component history is a design-evolution record rather than a collection of independent causal ablations. Future work will extend the loop with the SEv1.5 architecture, externally supplied MSA conservation scores, domain-specific objectives for catalytic activity, ligand binding and cofactor retention, and broader systems and longer physical trajectories, including the planned SPICE for Drug branch. The present manuscript focuses on the computational pattern and its released, auditable protein demonstration.

6. Data and code availability

The PDB/mmCIF-derived data, model and engine source, configurations, tests, logs, checkpoints, candidate records and pseudo-labels are released through versioned project archives. The SPICE model, engine and dataset-pipeline software are archived on Zenodo [20–22]. The processed protein corpus and released model data are archived on Hugging Face with immutable dataset revisions [23,24]. The released analysis artifacts support result-level re-analysis of the reported arrays and summary statistics. The CPU-bound RL/MD collection is represented by its released configurations, logs and derived artifacts. Supplementary Table S1a maps each headline result to its source artifact and analysis procedure; physical interpretation remains conditional on the documented model and protocol.

Artifact-to-claim map

Manuscript claim	Released artifact	Verification route
Native-label and simulator-accessible coverage	Pre-training corpus, <code>spice_protein</code> preprocessing and post-training environment configuration	Rebuild the corpus metadata and compare direct environment fields with the configured pH–temperature window
Coarse structural prior	Pre-training checkpoints, validation logs and CASP/contact evaluation scripts	Re-run the released pre-training evaluation
Path-A environmental exploration	<code>coverage.csv</code> , phase-map outputs and post-training logs	Recompute the tested environment envelope and collapse records
Path-B candidate admission	<code>pathb_candidates.csv</code> , failure records and survivor <code>.npz</code> files	Recompute survival, Q and physical-observable summaries using the stated Q-gate
Simulator-derived pseudo-label reflow	Pseudo-label <code>.npz</code> /TFRecord files, fine-tuning configuration and checkpoints	Rebuild the reflow stream and compare pre-training and reflowed metrics
Physical-engine implementation	<code>spice_engine</code> , modified dynamics and SPME dependencies, source tests and build files	Build the Rust/PyO3 components and run the released smoke and integration tests
Prior-scale comparison and published-data anchoring	<code>n10/n45000</code> run artifacts, comparison files, logs and analysis scripts	Reproduce the reported counts, Q summaries and qualitative comparisons from the released records

Compute footprint

Component	Resource	Wall-clock time
Pre-training, 45,000 chains	Kaggle T4 GPU	~2.5 h
Pre-training, 10 chains	Kaggle T4 GPU	~0.5 h
Post-training, N = 45,000	SLURM, 2 CPU cores and 6 GB RAM	1 d 3 h 13 m
Post-training, N = 10	SLURM, 2 CPU cores and 6 GB RAM	1 d 7 h 17 m

References

1. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
2. Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630**, 493–500 (2024).
3. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
4. Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv* (2021).
5. Leman, J. K. *et al.* Macromolecular modeling and design in Rosetta. *Nat. Methods* **17**, 665–680 (2020).
6. Valentini, A. *et al.* Polyformer: a generative framework for thermodynamic modelling of polymeric molecules. *arXiv preprint arXiv:2604.14241* (2026).
7. Jiang, F. *et al.* Pro-PRIME: A general Temperature-Guided Language model to engineer enhanced Stability and Activity in Proteins. *arXiv preprint arXiv:2307.12682* (2023; revised 2024).
8. Muellers, S. N., Allen, K. N. & Whitty, A. M. MEnTaT: A machine-learning approach for the identification of mutations to increase protein stability. *Proc. Natl Acad. Sci. USA* **120**, e2309884120 (2023). doi:10.1073/pnas.2309884120.
9. Jing, B., Berger, B. & Jaakkola, T. AlphaFold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845* (2024).
10. Swails, J. M., York, D. M. & Roitberg, A. E. Constant pH replica exchange molecular dynamics in explicit solvent using discrete protonation states: implementation, testing, and validation. *J. Chem. Theory Comput.* **10**, 1341–1352 (2014). doi:10.1021/ct401042b.
11. Shamsi, Z., Cheng, K. J. & Shukla, D. Reinforcement learning based adaptive sampling: REAPing rewards by exploring protein conformational landscapes. *J. Phys. Chem. B* **122**, 8386–8395 (2018). doi:10.1021/acs.jpcc.8b06521.

12. Wang, Z. *et al.* ProteinZero: self-improving protein generation via online reinforcement learning. *arXiv:2506.07459* (2025).
13. Tian, C. *et al.* ff19SB: Amino-acid-specific protein backbone parameters trained against quantum mechanics energy surfaces in solution. *J. Chem. Theory Comput.* **16**, 528–552 (2020). doi:10.1021/acs.jctc.9b00591.
14. Izadi, S., Anandakrishnan, R. & Onufriev, A. V. Building water models: a different approach. *J. Phys. Chem. Lett.* **5**, 3863–3871 (2014). doi:10.1021/jz501780a.
15. Essmann, U., Perera, L., Berkowitz, M. L., Darden, T., Lee, H. & Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **103**, 8577–8593 (1995). doi:10.1063/1.470117.
16. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *Proc. 35th Int. Conf. Mach. Learn.*, PMLR **80**, 1861–1870 (2018).
17. O'Connor, D. *dynamics*: a molecular-dynamics engine in Rust. GitHub repository (MIT licence). <https://github.com/David-OConnor/dynamics>
18. Lafaye, C. *et al.* Riboflavin-binding proteins for singlet oxygen production. *Photochem. Photobiol. Sci.* **21**, 1545–1555 (2022). doi:10.1007/s43630-021-00156-1.
19. Ranbhor, R., Kumar, A., Patel, K., Ramakrishnan, V. & Durani, S. Automated design evolution of stereochemically randomized protein foldamers. *Phys. Biol.* **15**, 036001 (2018). doi:10.1088/1478-3975/aaac9a.
20. SPICE-Protein. *spice_model: SPICE Model v1.0.0 with SOUP*, version v1.0.0 [Computer software]. Zenodo (2026). <https://doi.org/10.5281/zenodo.22274534>.
21. SPICE-Protein. *spice_engine: Release for Zenodo*, version v1.0.0_for_Zenodo [Computer software]. Zenodo (2026). <https://doi.org/10.5281/zenodo.22274612>.
22. SPICE-Protein. *spice_protein: Dataset Pipeline v1.0.0*, version v1.0.0 [Computer software]. Zenodo (2026). <https://doi.org/10.5281/zenodo.22275555>.
23. SPICE Protein. *spice_protein*, revision 98ef17f [Dataset]. Hugging Face (2026). <https://doi.org/10.57967/hf/9890>.
24. SPICE Protein. *spice_data*, revision bc1356d [Model]. Hugging Face (2026). <https://doi.org/10.57967/hf/10278>.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryMethodsfinal.docx](#)