

A Reinforcement Learning Approach to Protein Loop Modeling

Kevin Molloy^{ab}, Nicolas Buhours^{abc}, Marc Vaisset^{ab}, Thierry Siméon^{ab}, Étienne Ferré^c, Juan Cortés^{ab}

^aCNRS LAAS; ^bUniv de Toulouse LAAS; ^cSiemens Industry Software

Corresponding author: jcortes@laas.fr

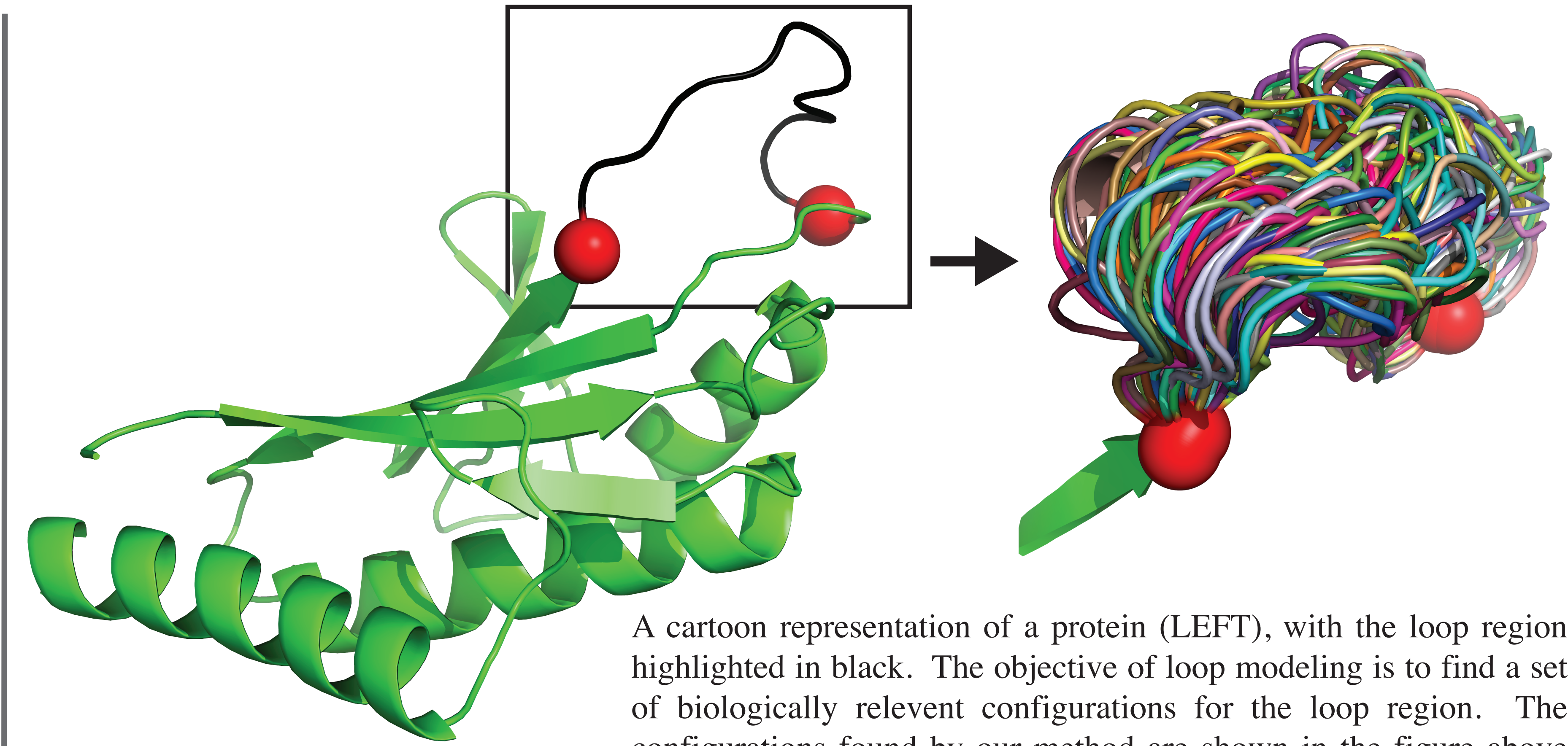


Objective

Our goal is to construct an ensemble of valid loop configurations for a protein system. Loops are flexible portions of proteins that control how the protein interacts with other molecular partners. We conduct an extensive search to effectively map the feasible conformation space for the loop region. This provides insight into the dynamics and energy surface associated with the loop configurations, which can be utilized to better understand and tune their functional roles.

Challenges

Modeling a protein's loop region presents many challenges. These systems exhibit high structural flexibility coupled with a large number of degrees of freedom (DOFs). This results in a vast search space. This high dimensional space is accompanied by a complex (non-linear, non-convex) conformational energy surface, which results in a rugged landscape with many local minima. As a result, many approaches are not applicable to solve this problem.



A cartoon representation of a protein (LEFT), with the loop region highlighted in black. The objective of loop modeling is to find a set of biologically relevant configurations for the loop region. The configurations found by our method are shown in the figure above (red spheres highlight the anchors of the loop).

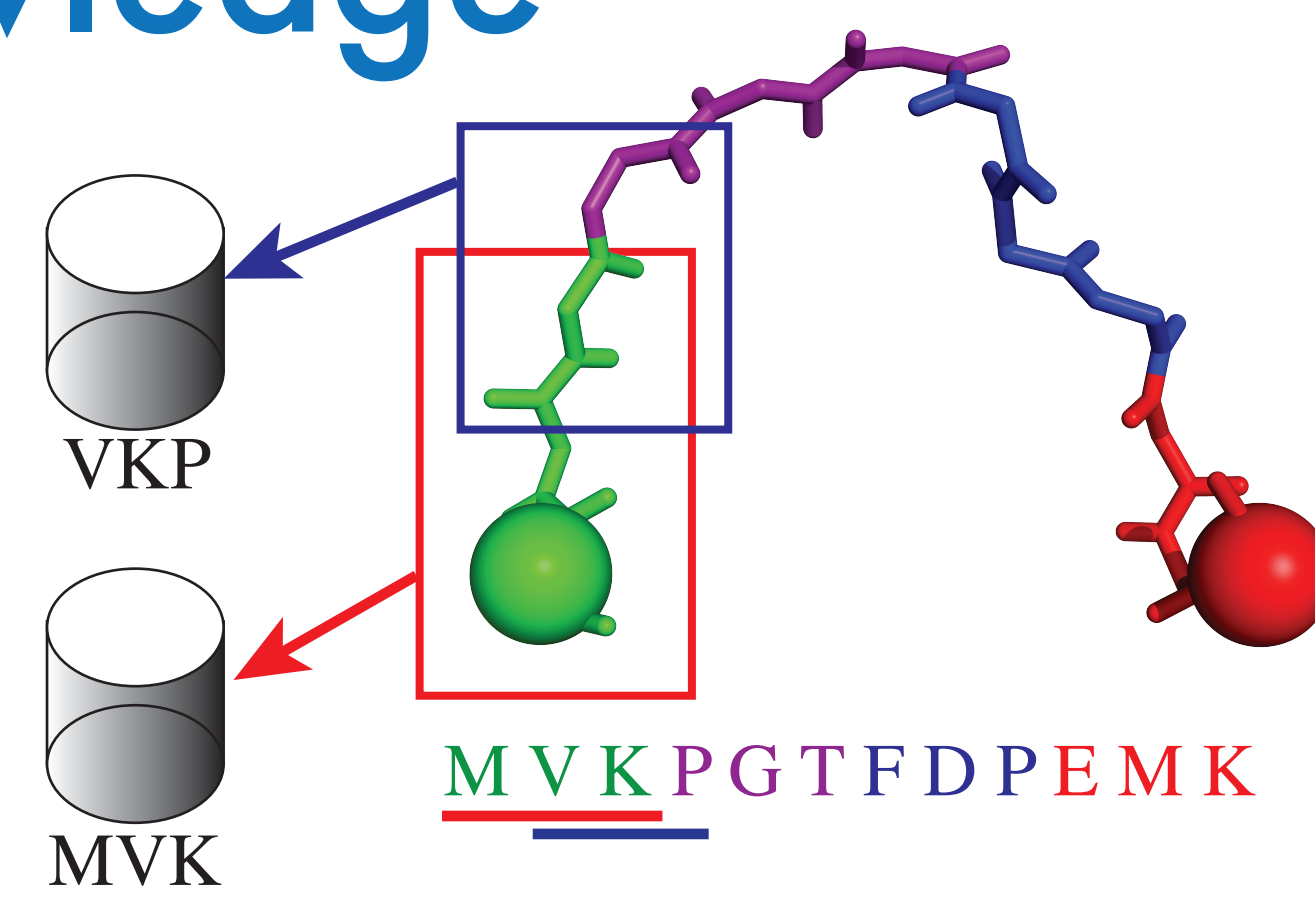
Mechanistic Model

Proteins are made of a chain of amino-acids (AA), which share a common scaffold (backbone) structure. A coarse-grained protein model allows for 2 DOFs per AA. The loop region of interest is modeled as a set of tripeptides (3 consecutive AAs). Each tripeptide has 6 dihedral DOFs, allowing for inverse kinematic (IK) solvers to be employed, treating the loop as a kinematic linkage¹. The figure below and right shows each tripeptide in a different color.

Prior Knowledge

The range of valid dihedral angles is limited. This range is learned by building a database (DB) of 1.1 million tripeptides from a set of 10,000 experimentally determined proteins². The figure on the right illustrates this process. A sliding window is passed over the AA sequence, storing each resulting tripeptide (organized by their AA ids) in the DB.

This approach, which has been used in other work, utilizes the DB to sample configurations for the tripeptides (see algorithm below).



Loop Construction

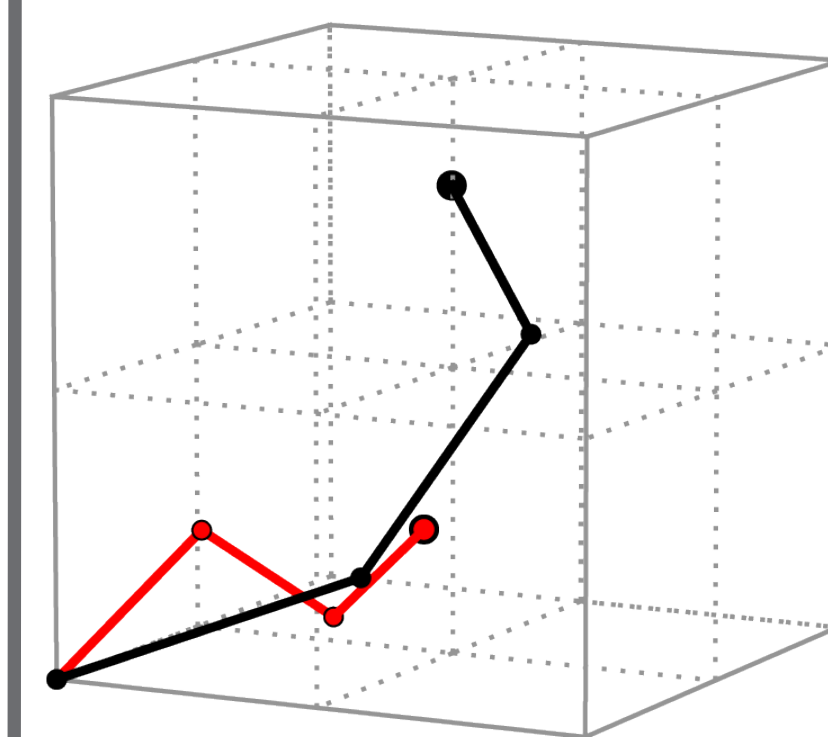
The algorithm constructs the loop by first selecting a tripeptide, S (loop consists of K tripeptides) to solve via IK. The loop is incrementally constructed by sampling tripeptides from the database for positions 1 to S - 1, and then from position K to S + 1, and finally solving for tripeptide S. A tripeptide sample is accepted if it is collision free. If after sampling MAXTRY times we fail to locate a valid sample, the algorithm backtracks to the prior tripeptide and resamples.

```
bool placeTripeptide (loopPlan, position)
if (position == lastPosition)
return(solveIK(loopPlan));
else {
for (attempts= 0;attempts< MAXTRY;++attempts) {
tripeptide = SampleTripeptide (loopPlan, position);
InstallTripeptide (loopPlan, currPosition, tripeptide);
if (CollisionFree(loopPlan,position) {
success = placeTripeptide (loopPlan,position+ 1)
if (success) break;
} else success = false;
}
return(success);
```

PSEUDOCODE

Methods

Learning how to Select Tripeptides

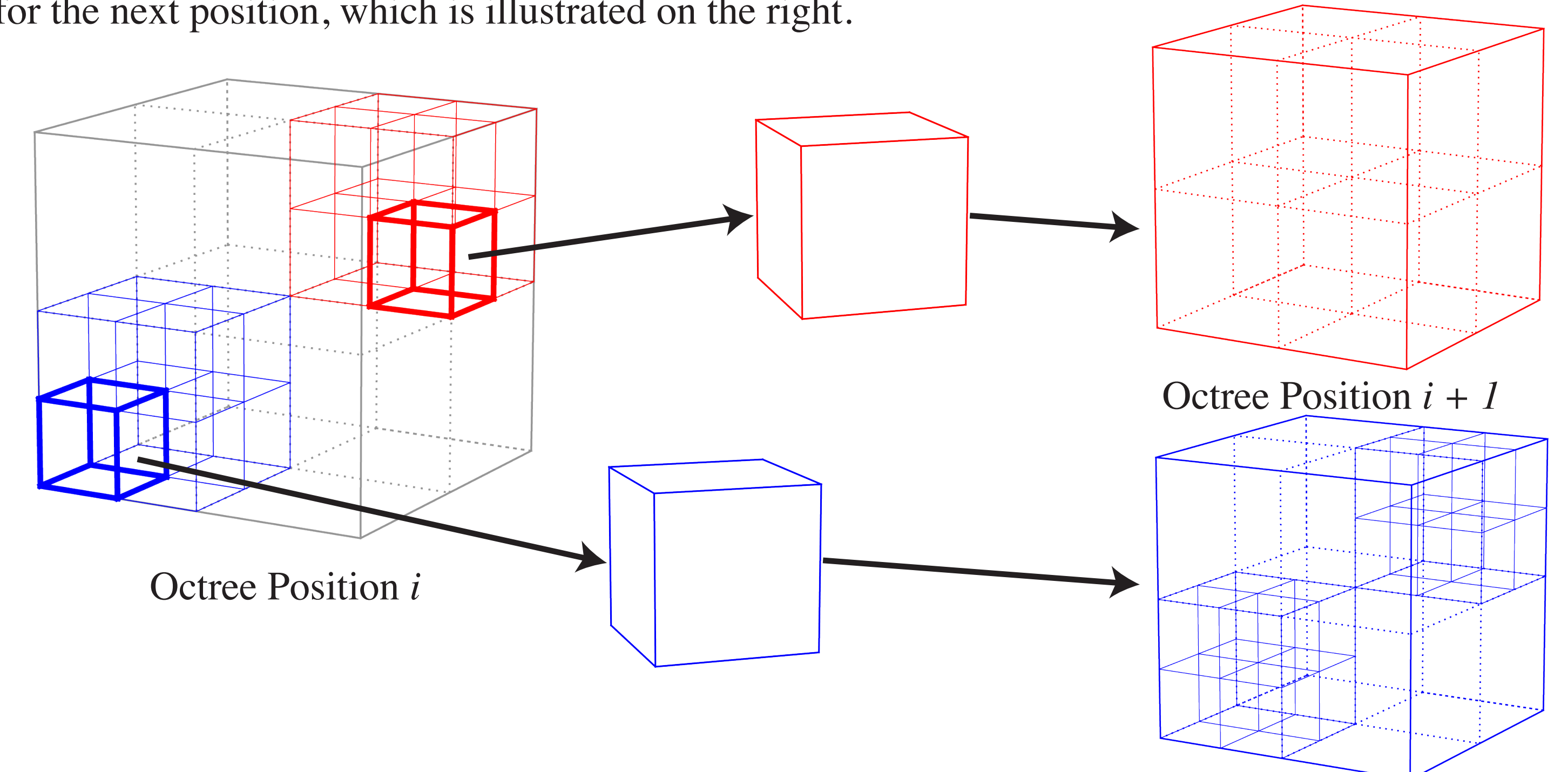


An octree organized by end effector position.

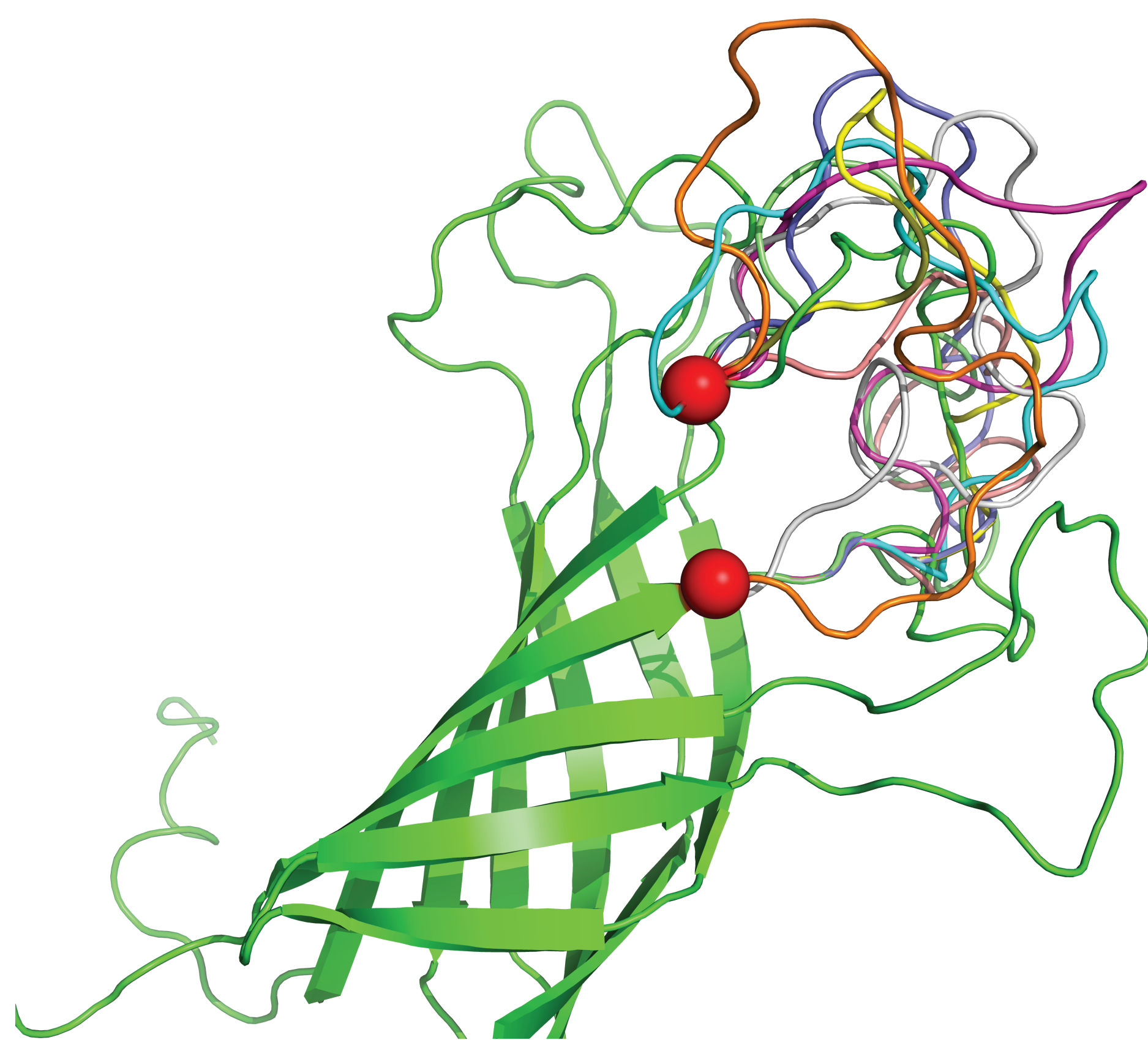
When constructing a loop, we select a tripeptide from the subset containing the same 3 letter AA code. But how do we select a tripeptide from this subset? Our method utilizes a reinforcement learning (RL) approach³. First, the subset of tripeptides for each position are organized into an octree (or higher dimensional object) by their features. Cells in the octree are rewarded if they participate in successful loop closures. Several sets of features for projecting the tripeptides into the tree are investigated:

- Length
- End effector position
- Orientation and length

We investigate reward mechanisms (recording successes or failures) and selection schemes (including greedy and probabilistic). The success of placing a tripeptide k is dependent on the previous $k-1$ selections. This dependency is captured by having each grid cell point to an entirely new tree structure for the next position, which is illustrated on the right.



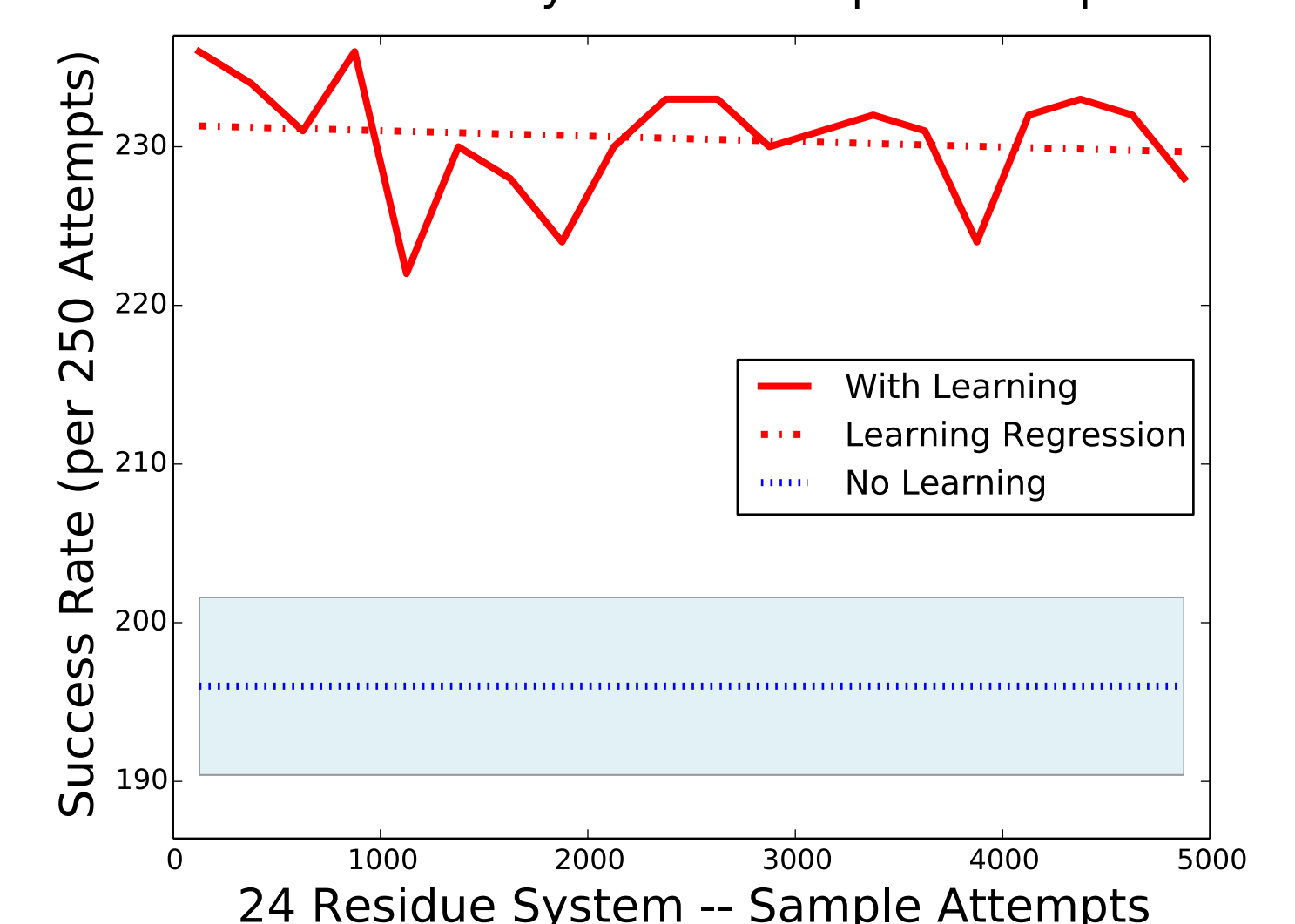
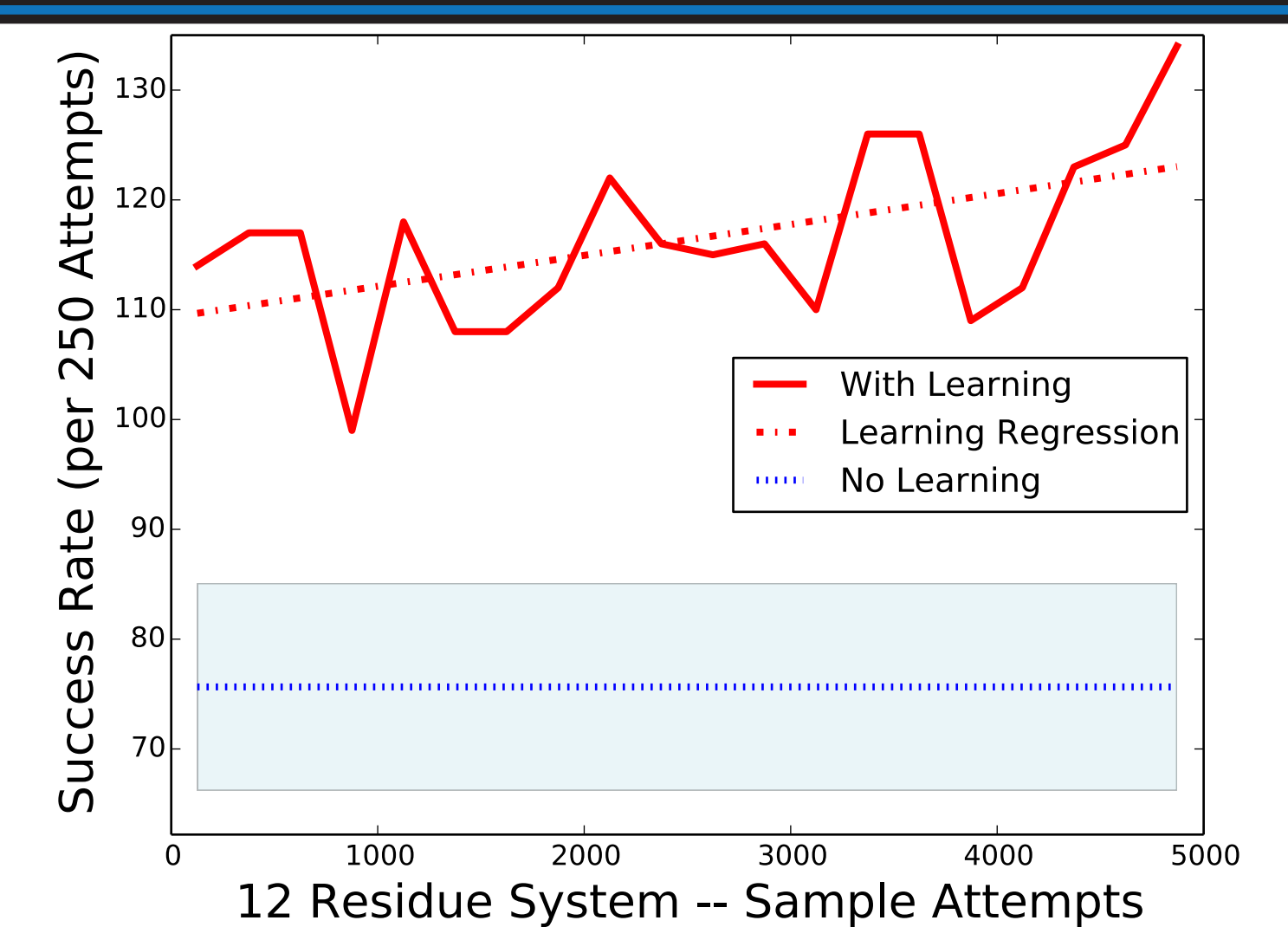
Results



We tested this approach with and without the learning component on several proteins and showcase two of them, with loop lengths of 12 and 24 residues. When the learning component is excluded, tripeptides are selected from the database uniformly at random. Each loop takes on average 4 and 5 seconds to compute for the 12 and 24 residue systems respectively with the learning agent.

The figure (top right) shows the rate that loops are successfully generated (12 residue system w/greedy selection strategy). The solid red lines indicates the number of successful loops generated over the last 250 attempts when the learning agent is active, the dash red line shows the average success rate, the blue dotted line shows the average success rate when the learning agent is inactive, and the blue area around this line shows the standard deviation. Clearly the learning agent improves the success rate.

The figure (left) illustrates a few loops for the larger system (24 residues) and the corresponding plot (bottom right) illustrates the performance of the learning agent. The learning rate is slower for the larger system due to the higher dimensionality of the search space.



References

Loop modeling is an important open problem in the field of structural biology. Computational methods, like the one proposed here, supplement available experimental data and augment our understanding of the role of protein loops. This technique of applying prior knowledge and learning can be adapted to augment robotic motion planning problems that require the generation of samples in high dimensional space.

Conclusion

- 1 J. Cortés, T. Siméon, M. Remaud-Simon, and V. Tran. "Geometric algorithms for the conformational analysis of long protein loops", in *Journal of Computational Chemistry*, 2004.
- 2 R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1998.
- 3 A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. "SCOP: a structural classification of proteins database for the investigation of sequences and structures", 1995.