Gradient Descent Optimization and Deep Reinforcement Learning for Protein-Protein Interaction

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Master Science Computer Science

by

ELHAM SOLTANI KAZEMI

Dr. Jianlin Cheng, Thesis Supervisor

DECEMBER 2022

The undersigned, appointed by the dean of the Graduate School, have examined the Thesis entitled

Gradient Descent Optimization and Deep Reinforcement
Learning for Protein-Protein Interaction

presented by Elham Soltani Kazemi,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Jianlin Cheng

Toni Kazic

K S M Tozammel Hossain

Derek Anderson

Acknowledgements

I would like to express my sincere gratitude and appreciation to my adviser, Dr. Cheng, for his encouragement and support over the two years. I would also like to thank committee members, Dr. Anderson, Dr. Duan, Dr. Kazic, and Dr. Hossain, for serving on my committee.

Many thanks go to the members of the Bioinformatics and Machine Learning Lab for rewarding research discussions. Special thanks go to Farhan Quadir, Raj Roy, Nabin Giri, Ashwin Dhakal, and Sajid Mahmud for their assistance and guidance in many matters. Thanks are also due to Frimpong Boadu for proofreading some of this thesis.

Table of Contents

Abstract

Reconstruction of the 3D structure of protein dimers is a crucial and challenging task. Although inter-protein contacts have been found useful in the modeling process of protein complexes, a few methods have been introduced to tackle the challenging quaternary structure prediction problem utilizing inter-chain contacts. We propose an optimization method based on gradient descent algorithm, called GD, to reconstruct the quaternary structures of protein complexes from inter-protein contacts. We test the performance of the GD method on both homodimers and heterodimers utilizing both true and predicted inter-protein contacts. GD has a superior performance than a Markov Chain Monte Carlo (MC), and a method based on Crystallography and NMR System (CNS). When native inter-chain contacts are provided as inputs, GD builds high quality models with TM-scores of more than 0.92 and interface RMSDs (I_RMSDs) of less than 1.64 Å for both homodimers and heterodimers. Receiving the predicted inter-chain contacts as restraints, GD is able to generate models with a mean TM-score of 0.76 for 115 homodimers. Besides, for nearly half of the homodimers, GD reconstructs high quality models with TM-scores more than 0.9 using just the predicted inter-chain contacts to guide the modeling process.

We also develop a self-learning algorithm based on reinforcement learning, named DRLComplex, to reconstruct protein dimers from true/predicted inter-protein contacts. We evaluate DRLComplex on two standard datasets including CASP-CAPRI dataste (28 homodimers), and Std32 (32 heterodimers). If native inter-chain contacts are provided, DRLComplex generates models with mean TM-score of 0.9895 and mean I_RMSD of 0.2197 for CASP-CAPRI dataset, and models having average TM-score of 0.9881, and average I_RMSD of 0.92 for Std32. Using predicted inter-chain contacts as restraints, DRLComplex builds models with overall average TM-scores of 0.73 and 0.76 for CASP-CvAPRI and Std32, successively. Moreover, utilizing predicted contacts, DRLComplex improves the mean I_RMSD of the reconstructed models for the Std32 dataset by 0.29%, 1.01%, 13.47%, and 8.69% over GD, MC, CNS, and Equidock (an end-to-end quaternary structure prediction method), respectively. In addition, the mean I_RMSD of the models predicted by DRLComplex for CASP-CAPRI dataset utilizing predicted contacts is 0.04, 3.94, and 4.07 lower than MC, CNS, and Equidock.

Codes for GD, DRLComplex, and GD for multimers are available at https://github.com/jianlin-cheng/DeepComplex2.git, https://github.com/jianlin-cheng/DRLComplex.git, and https://github.com/BioinfoMachineLearning/GD-multimer.git, respectively.

Introduction

Proteins are essential in the day-to-day lives of humans; they form a fundamental part of the human body and have several vital roles for the normal functioning of the human body. Amino acids are the building blocks of proteins, they bond with each other, to form different types of proteins. There are 20 different amino acids which are combined to form different proteins. Every combination of these amino acids produces a different kind of protein.

There are four types of protein structure, the primary structure, secondary structure, tertiary structure, and quaternary structure. The primary structure of a protein refers to the sequence of amino acids, held together by peptide bonds to form a polypeptide chain. The secondary structure refers to the local substructure on the polypeptide backbone chain. The tertiary structure is the 3D structure, created by a single protein molecule and the quaternary structure refers to the 3D structure formed by two or more proteins, operating together as one functional unit. The quaternary structure is also referred to as protein complex, where two or more complexes are known as multimer.

The quaternary structure of proteins is very important since it is closely related to its function, cellular processes and also play significant roles in designing and discovering new drugs [1, 2]. There are few protein complexes known, and identifying them through biological experimentation is very expensive, however, there is numerous known amounts of protein data available on other protein structure, interactions within proteins and among proteins of similar amino acid combinations. Research has shown that the quaternary structure of proteins is related to the inter-chain contacts between two or more proteins and also interactions between proteins is usually correlated especially when the involving proteins have similar primary structure or tertiary structure. This makes it necessary to explore the available data to create models that can be used to predict the quaternary of proteins as efficiently and accurately as possible.

There are several computational methods developed that leverage on this available protein data [3-11]. One of the most widely used approaches for modeling complex structures is Computational protein docking, the approach takes the tertiary structures of individual proteins as input to build the quaternary structure of the complex as output. Docking methods can be largely divided into two categories including template-based modeling, in which known protein complex structures in the Protein Data Bank (PDB) are used as templates [10-17] to guide modeling, and template-free modeling (ab initio docking), which does not use any known structure as template, and instead searches through a large conformation space for relative orientations of protein chains with minimum binding energy. The binding energy is often roughly approximated by geometric and electrostatic complementarity, inter-chain hydrogen binding, hydrophobic interactions, and residue–residue contact potentials [18-23]. Another method widely used is the Ab initio docking, which is suitable for protein complexes that lack suitable templates. However, ab initio docking methods need to search through a huge conformation space, which is usually not feasible with limited time and computing resources [23-30].

Gradient descent optimization has become a popular method to build the tertiary structure of proteins using intra-protein (intra-chain) residue–residue contacts or distances and they have shown impressive results [31, 32].

In this work, we present two methods for constructing the quaternary structure of proteins. The first method is a distance-based reconstruction from the inter-chain contacts, and the second is a reconstruction of quaternary structures using deep reinforcement learning.

In the first method, we use gradient descent optimization to build quaternary structures of protein dimers using the inter-chain contacts as contact/distant constraints. Our algorithm works by randomly initializing an arbitrary quaternary structure from tertiary structures of protein chains and combining with true inter-chain contacts to reconstruct high-quality quaternary structure. The approach is evaluated on several datasets of homodimer and heterodimers, and it performs better than simulated annealing and Markov chain monte Carlo simulation methods.

The second method is an agent-based self-learning deep reinforcement learning method. Here, we use a reinforcement learning approach unlike the first one which uses stochastic gradient descent. We test this method on two standard datasets of homodimer and heterodimer (the CASP-CAPRI homodimer dataset and Std32 heterodimer dataset).

Optimization Methods in Theory

Cost Function & Gradient Descent

Considering an input dataset $\{(x^{(i)}, t^{(i)})\}_{i=1}^{N}$, where $x^{(i)}$ represents the features at time i, and $t^{(i)}$ is the true labels at time i. There is a function $t = f(x)$, it is this function that we seek to approximate through learning. The learning model gives a function $y = f(w, x)$, where y is the predicted labels of the model. We use a cost function $L(y, \hat{y})$ to determine the magnitude of difference between these two functions.

The cost function tells us how well the model fits the data. We need to find a low cost, which indicates that the predicted function approximates the actual function well. Given the cost function as $J(\theta)$, our goal is to find θ, such that $J(\theta) = 0$. We may not find the value of θ for which $J(\theta) = 0$, but may find a good estimate. Therefore, we seek to minimize the value of $J(\theta)$ as much as possible.

Gradient descent is an optimization technique, used to find the parameters of a function that minimizes the cost function. Some functions are such that we can compute these parameters analytically or in one step. Gradient descent, however, is useful when analytical computation of these parameters is not feasible. We usually use gradient descent to minimize the cost function described above.

Simply, gradient descent works as follows; initializing the parameters of the function and calculating the cost associated with them. The derivative of the cost is computed and used to update the parameters. The process is repeated until convergence, or the cost is close enough to zero.

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta)$$

Equation 1

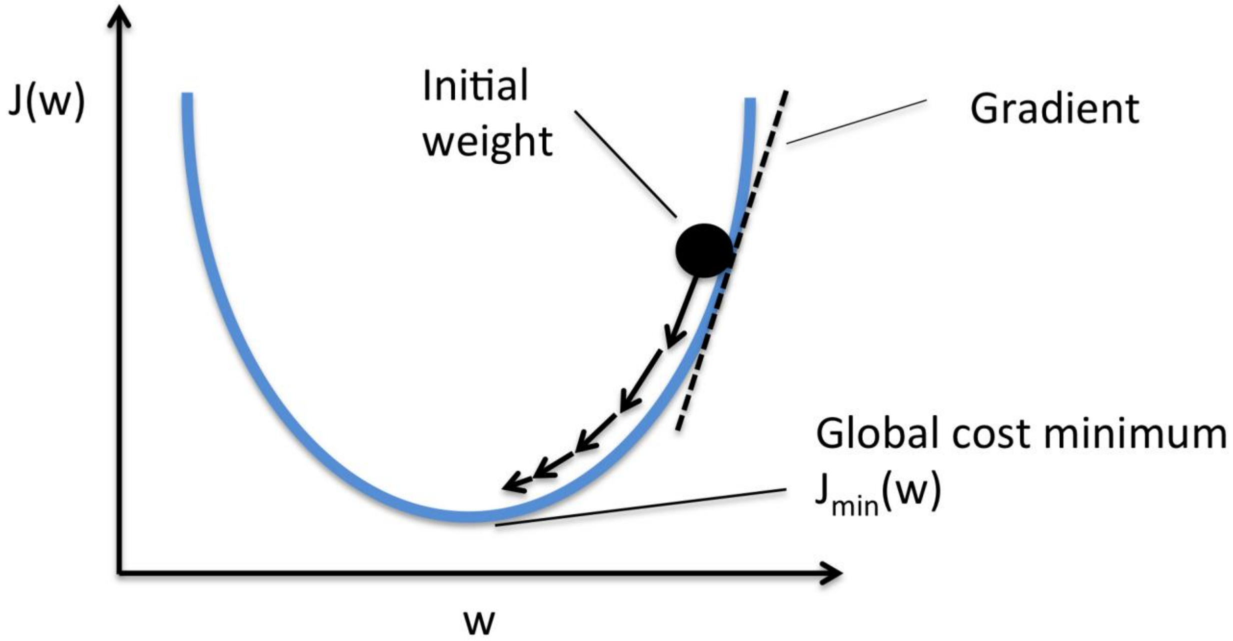

Figure 1. The above image shows the behavior of gradient descent for a 1-dimensional space. The intuition is that the sign of the gradient points us in the direction to move. From the diagram, moving down the slope, reduces the corresponding value for $J(w)$. If we the function is convex and we choose an appropriate step size, as well as move in the direction that reduces $J(w)$, then we are guaranteed that it will converge. The above scenario shows the behavior of gradient descent for $1-$dimensional space, which is easier to visualize.

Gradient Descent & Taylor Series

To explain further, consider the k^{th}-order Taylor approximation of f at x.

$$f(x+\triangle x) \approx f(x) + f'(x)\triangle x + f''(x)\frac{(\triangle x)^2}{2!} + ... + f^{(k)}(x)\frac{(\triangle x)^k}{k!}$$ Equation 2

The Taylor series tells us that, if we have a function $f(x)$, and we know the value of the function at a point x, then we can estimate the value of the function at a new point $x + \triangle x$, Where $x + \triangle x$ is very close to x.

From equation above, the value of $f(x+\triangle x)$ depends on $f(x)$(first term) and $\triangle x$(other terms). If $\triangle x$ is such the sum of other terms is negative, then $f(x+\triangle x) < f(x)$. That is an approximation of a point very close to x on the function, and also this new point reduces the value of $f(x)$.

The gradient descent uses the first order derivative of the Taylor series and hence we have:

$$L(\theta + \eta u) = L(\theta) + \eta * u^T \bigtriangledown_\theta L(\theta)$$

This means that if the move ηu reduces the loss then $\eta * u^T \bigtriangledown_\theta L(\theta) < 0$.

In using gradient descent algorithm, if the function is convex, then we have a global minima, however if the function is not convex, then we may have several local minima. Depending on our choice of the learning rate and depending on how well conditioned the problem is, we may end up with one of many solutions. A very small learning rate will cause Θ to be slowly updated and will require many iterations for a better solution. A very large learning rate will cause undesirable divergent behavior in the learning process.

Gradient Descent Algorithm

1. choose initial point $\theta^{(0)} \in \mathbb{R}^n$
2. $\theta^{(k)} = \theta^{(k-1)} - t^k \cdot \triangle f(\theta^{(k-1)})$
3. Repeat for $k = 1, 2, 3, \ldots$
4. Stop at some point.

Stochastic Gradient Descent

The gradient descent algorithm is computationally expensive. In stochastic gradient descent, a uniform sample of the data is used to update the parameters at each iteration. This reduces the computation enormously.

As the algorithm sweeps through the training set, it performs the above update for each training sample. Several passes can be made over the training set until the algorithm converges.

Stochastic gradient Descent Algorithm

- Choose an initial vector of parameters θ and learning rate η.
- Repeat until an approximate minimum is obtained:
    a. Randomly shuffle samples in the training set.
    b. For $i = 1, 2, 3, \ldots$ do:
        i. $\theta = \theta - \eta \bigtriangledown f_i(\theta)$

Deep Reinforcement Learning

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error. In summary, reinforcement learning is learning by trial-and-error, that is learning solely from rewards or punishments. We define some terms associated with reinforcement learning below:

- Agent: The agent is responsible for taking actions. In this case, the agent is the algorithm.
- Environment: The environment is the world or context through which the agent moves, and which gives feedback to the agent.
- State: The state is the part of the environment or the situations which the agent finds can be in.
- Action: An action refers to the steps or decisions the agent can take.
- Reward: In reinforcement learning, the reward is the feedback which measures the success or failure of an agent's action at a given state.
- Policy: The policy is the strategy which the agent employs to determine the next action based on the current state. It maps states to actions, the actions that promise the highest reward.
- Value: The value is defined as the expected long-term reward.

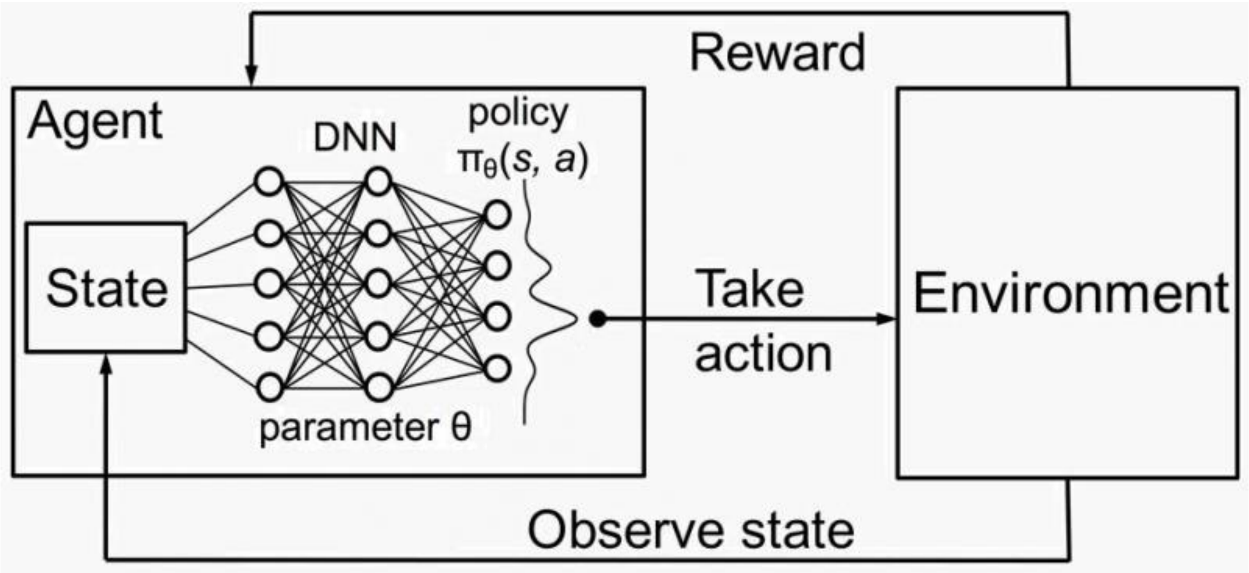

Figure 2. Agent interacting with the environment.

Neural networks are used to approximate functions, they are used to recognize underlying relationships in a set of data through learning. They are useful in reinforcement learning, especially when the state space or action space are too large to be completely known.

Applying neural nets to reinforcement learning is known as deep reinforcement learning. Here, the agents construct and learn their own knowledge directly from raw inputs, such as vision, without any hand-engineered features or domain heuristics. Essentially, the neural network is used to approximate a value function, or a policy function. The neural network can be trained to learn a function which learns a mapping of states to values, or state-action pairs to Q values, eliminating the need for a lookup table, which stores, index and update of all possible states and their values. This is important because lookup tables may not be feasible for large problems.

Neural networks learn parameters that approximate the function relating inputs to outputs, similar to every other neural network.

Methods

Developing robust optimization methods to reconstruct protein structures using residue-residue distances

In 2015, [33] introduced a contact-based *ab initio* structure modeling approach named CONFOLD, which generates protein structures using predicted distance restraints and secondary structures. CONFOLD Transforms predicted contacts into C_b-C_b distance constraints and predicted secondary structures into distance and dihedral angle constraints. A spatial optimization algorithm then combines distance geometry information with simulated annealing to reconstruct three-dimensional structure of proteins in a way that distance constraints are satisfied as best as possible. Unlike Crystallography and NMR systems (CNS) [34] which uses distance geometry constraints to generate 3D structure of proteins in one shot, CONFOLD has an extra stage of removing noisy, and physically unsatisfied constraints and creating new constraints (especially for beta-strands generated in the first step) to enhance strand pairings. Besides, CONFOLD augments its contact/distance constraints with the help of constraints extracted from predicted secondary structures. It must be noted that these additional constraints are not used by CNS; hence, CONFOLD is able to reconstruct models of higher quality (with more accurate secondary structures) compared to CNS. Also, CONFOLD builds models from contacts that are much more accurate than template-based modeling tools such as Modeller [35].

Another modeling method is Rosetta, which is a fragment-assembly method for building protein structures. The main difference between CONFOLD and Rosetta [36] is that the former directly converts predicted contact restraints into 3D structure of the desired protein, thus, the restraints have a direct and major role in the *ab initio* modeling process, whereas the latter uses the contact restraints as part of its complex energy function to conduct the fragment-assembly of protein structures and therefore the contact restraints participate indirectly in modeling process. Fragment assembly utilizes additional fragment knowledge to guide the modeling process; however, the extra information is known to be more suitable for small proteins having uncomplicated structure. The process mostly relies on disconnected, random fragment assembly, as a result, most of the time, it fails to build a model close to the true structure, especially for large proteins having complex structure, even if the provided contact constraints are of high quality.

Although existing distanced-based 3D modeling methods (e.g., CONFOLD and Rosetta) achieved promising results for some proteins, they often fail to reach near native conformation if the provided contacts/distance restraints are noisy and inaccurate. Furthermore, using the whole restraints to build the entire structure at once will make the modeling process of large proteins so complicated. Apart from the complexity issue, conflicting restraints that cannot be satisfied simultaneously might puzzle and mislead the modeling process. Thus, it seems crucial to handle inaccurate restraints, the complicated modeling process, and contradictory restraints in order to be able to reconstruct protein structures when provided restraints are noisy and contain inconsistent information.

To solve these problems, we implemented a robust gradient descent method, the most widely used optimization algorithm for deep neural networks, to build protein structures using distance constraints. Besides, to reduce the complexity of the modeling process and handle inaccurate and contradictory restraints, we carefully controlled the amount and order of the restraints passed to the optimization process. We believe that different proteins having different structural complexity must have different modeling process, hence, we designed three different approaches to feed distance restraints into the folding process: (1) passing the constraints from short to medium to long range; (2) adding distance information batch by batch in a stochastic way; (3) adding the whole distance restraints to the optimization process at once. Our simple gradient descent algorithm receives true residue-residue distances, and secondary structure information to adjust the position of x, y, and z coordinates of the conformation in a way that the error function is minimized. The error function (cost function), and its derivative with respect to the given distance are shown in the following:

$$p = \frac{1}{\sigma\sqrt{2\pi}}\ exp\left[-\frac{1}{2}\left(\frac{distance}{\sigma}\right)^{2}\right]$$

$$error\ function\ =\ -ln\,p\ =\ -\frac{1}{2}(\frac{distance}{\sigma})^{2}\ -\ ln\ \frac{1}{\sigma\sqrt{2\pi}}$$

$$\frac{d\ error - function}{d\ distance} = -\frac{distance}{\sigma}$$

The folding process often fails to correctly model protein structures if all true restraints are used at once. This happens because the above cost function is nonconvex and therefore simultaneous optimization of all distance restraints may result in a bad local minima. As mentioned above, adding the distance restraints batch by batch or in a hierarchical way (from short to medium to long range) can help the optimization process to correctly reconstruct topologies of some proteins. However, the folding process still gets stuck into bad local minimums for some proteins having complicated topologies and a lot of restraints. In another attempt to tackle the local minima issue, we developed a repetitive gradient descent algorithm. Since the initial start affects the optimal solution, we separately started our optimization algorithm from 30 different random initial conformation. The algorithm outputs the conformation having the lowest energy function as the final model (see Figure 3). We believe that starting from different initial starts can help the optimization process to escape bad local minimas and generates higher quality conformations.

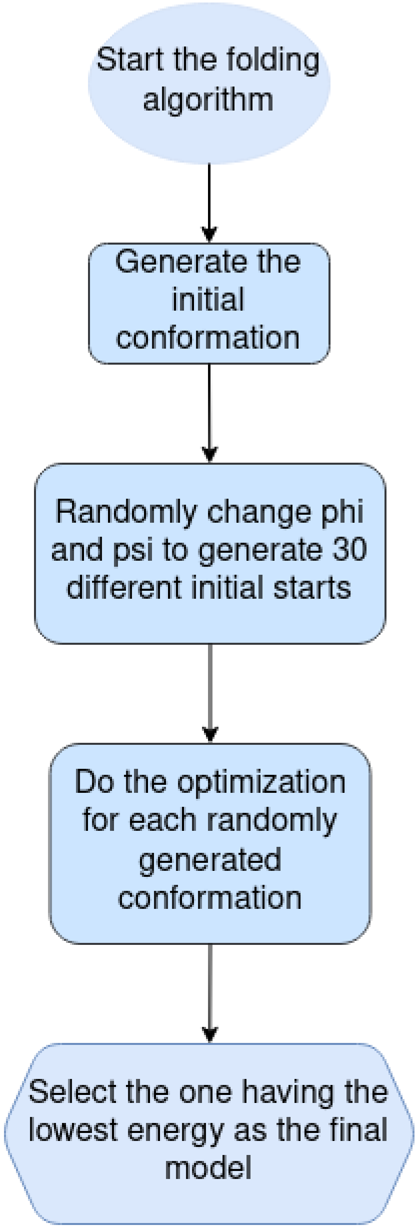
Start the folding algorithm
Generate the initial conformation
Randomly change phi and psi to generate 30 different initial starts
Do the optimization for each randomly generated conformation
Select the one having the lowest energy as the final model

Figure 3. Repetitive gradient descent algorithm.

Gradient Descent Method:

Gradient descent utilizes inter-chain contact/distance restraints to reconstruct quaternary structures of protein complexes. The cost function to examine if any two residues form a contact in order to guide the modeling process is shown in the following equation.

$$f(x) = \begin{cases} \left(\frac{x-lb}{sd}\right)^2 & x < lb \\ 0 & lb \leq x \leq ub \\ \left(\frac{x-ub}{sd}\right)^2 & ub < x < ub + sd \\ \frac{1}{sd}\left(x - (ub + sd)\right) & x > ub + sd \end{cases}$$

lb and ub are the lower bound and upper bound of the distance between any two residues supposed to form a contact. Two residues are said to be in contact if the distance between their heavy atoms is less than or equal to 6Å. However, for simplicity, we assume two residues are in contact if the distance between their C_β atoms (C_α atoms for Glycine) is less than or equal to 6Å. Based on the cost function, if the distance between two residues is between lower bound (lb) and upper bound (ub), then the constraint is met, and consequently the error (cost) is 0. lower bound (lb) and upper bound (ub) are set to 0 and 6, respectively. sd (Standard deviation) is also set to 0.1.

The costs of all the contacts participating in the modeling process are added up into an objective function, named contact energy. For the sake of simplicity, we give identical weights to all the restraints to let them equally participate in the process. The contact energy is differentiable with respect to residue-residue distances, and x, y, and z coordinates of atoms, hence, it can be optimized with the help of a gradient descent-like algorithm (GD),e.g. Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS).

We use a stand-alone package, called PyRosetta, to implement GD. The total objective function to be optimized is the sum of the contact energy and talaris2013 potentials. Talaris2013 potentials have proven to be effective in improving the quality of the final model. The GD takes inter-protein contacts and an arbitrary initial conformation of a complex, assembled from the tertiary structures of protein partners, as its inputs. The tertiary structures might be native structures (in the bound state), or structures predicted by existing methods. Predicted monomer structures are assumed to be in the unbound state since they are mostly predicted without taking into account the other chain partners. To prepare the initial conformation, the monomer structures of the two protein partners undergo 40 random rotations and translations between 1° and 360°, and 1Å and 20Å, respectively.

More specifically, the protein partners are arbitrarily rotated and translated along their line of centers so as to face each other.

The gradient descent algorithm (e.g., LBFGS) outputs the final model after 6000 iterations (for some protein dimers, the algorithm converges after only 1000 iterations). As discussed earlier, a good initial start is needed to prevent the optimization algorithm from getting stuck into bad local minimums. Therefore, we perform multiple optimizations starting from several random initial conformations (in the hope to avoid falling into bad local minimums), the model having the lowest energy function is chosen as the final structure.

Markov Chain Monte Carlo Optimization:

We use a Rosetta protocol based on Metropolis-Hashing sampling to perform a Markov Chain Monte Carlo (MC) optimization to build dimer structures based on Boltzman distribution. An initial random conformation of a protein complex is prepared just as the GD algorithm. Then, one chain of the initial conformation is rotated and translated relative to the other chain to build a new structure in the MC optimization. Afterwards, 500 Monte Carlo moves are made, among them some are accepted or rejected according to the standard Metropolis acceptance criterion. This process is called low-resolution search.

Following the low-resolution search, a Newton optimization algorithm is used to further refine the conformation (e.g., back-bone and side chains); this process is named high-resolution refinement process, for which the conformation is rotated and translated along the direction of the gradients of the objective function, with the main purpose of detecting a conformation with the lowest energy function in the translation/rotation space. The high-resolution process is repeated several times until it finds a local minimum of the objective function that is as good as the global minimum.

As shown in Figure 4, MC method utilizes low-resolution search and high-resolution refinements implemented in RosettaDock from Pyrosetta to minimize the same objective function as the GD algorithm. Low-resolution and high-resolution docking are performed using the DockingLowRes protocol and DockMCMProtocol, respectively. For each protein complex, MC method performs 10^5 to 10^7 rounds of optimization using different random initial starts. Analogous to the GD algorithm, MC selects the model with the lowest energy function as the final model.

Simulated Annealing Optimization Based on Crystallography and NMR System (CNS):

This optimization approach, named Con_Complex in the DeepComplex package, utilizes a simulated annealing protocol from the Crystallography and NMR system (CNS) to reconstruct a protein complex by optimally satisfying given inter-protein contacts. The method receives the tertiary structures of protein chains of a protein multimer (e.g., homodimer, heterodimer) as well as true/predicted inter-chain contacts to build the multimer structure. It must be noted that the monomer structures remain unchanged throughout the process. Before starting the optimization, CNS converts contact restraints into distance restraints. The model builds 100 structural models and outputs the top 5 models having the lowest energy value. The CNS method is able to build the quaternary structures of multimers, regardless of having identical or different chains. Using CNS, the quality of the reconstruction model highly relies on the quality of the given contact restraints, mainly because contacts are the main restraints to dictate whether or not a particular model is of high quality.

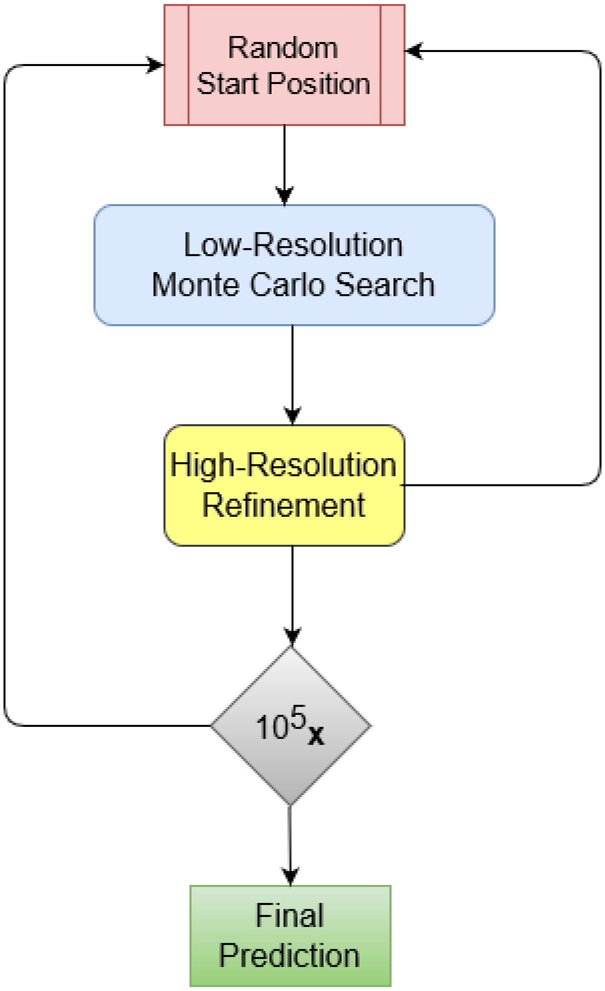

Figure 4. MC algorithm.

Deep Reinforcement Learning to Build Protein Complexes via Self-learning:

Here, we utilize a deep reinforcement learning approach, named DRLComplex, to reconstruct the structures of protein complexes through an automatic self-learning process that adjusts the position of one protein chain (ligand) relative to the other (receptor) to reach a native or near native conformation. More specifically, an artificial intelligence agent learns to interact with a modeling environment by selecting actions (from a set of legal actions) based on immediate reward and long-term reward to modify the structure of a protein complex (state of the environment). The modeling environment is implemented in the Pyrosetta package.

The state of the environment (S) is represented using the 3D structure (all atomic coordinates) of a given protein dimer as well as its $C_\alpha - C_\alpha$ inter-protein distances (distance Map). The 3D coordinates of atoms are useful to build new conformations by applying the chosen action to it, whereas the distance map is used by a deep learning model to compute Q-values for all possible actions. The distance map is represented by a two-dimensional vector of size $L_1 \times L_2$, where L_1 and L_2 define the lengths of the ligand and the receptor in a protein complex, respectively. The distance map assigns each cell with the distance between C_α atom of a residue in the ligand and C_α atom of a residue in the receptor. Possible actions for the agent to adjust the position of the ligand with respect to the receptor are as follow: three translations along x, y, and z-axes with a step size of 1Å, and three rotations about x, y, and z-axes of 1°. The Q-function of the reinforcement learning is represented by a deep convolutional neural network. As shown in Figure 5, a CNN takes a state S as input, and outputs the Q-values of six actions. At each time step, the agent calls this prediction network to compute Q-values corresponding to the six actions to determine the optimal action to modify the conformation (state) of a protein complex.

The CNN network is trained using a sequence of state (S), action (A), reward (R), and next state ($S^{'}$) gathered by the agent through a continuous interaction with the modeling environment to pick an action based on $\varepsilon - greedy$ strategy to adjust the position of the ligand. Following the $\varepsilon - greedy$ policy, in each episode, the agent decides to pick the action recommended by the prediction network with a probability of $1 - \varepsilon$, otherwise, it randomly selects one of the six actions. After the selection of an action (A), the environment (E) applies A to the 3D conformation of the current state (S) to build a new 3D conformation of the next state ($S^{'}$). The value of the immediate reward (R) is computed as the difference between the quality of the current state S and next state $S^{'}$ respecting to the reference state S^{*}, which might be the native conformation of a protein complex, or the predicted inter-proteins contacts passed to the environment at the beginning of episdoes. The agent continues this process in order to generate a sequence of states, actions, rewards, and next states. As can be seen in Figure 6, the CNN model is trained using agent's experiences (S, A, R, $S^{'}$) sequentially stored in a replay buffer over a number of time steps, such a process is called experience replay technique.

Given a state S and an action a, the reference Q-value, named $Q_{reference}$(S, a), is generated using the immediate reward and the output of a network called target network (Q_{target}). The reference Q-values are used as labels to guide the training process of the CNN model. Every k steps, the weights of the CNN model are assigned to the weights of the target network to calculate the Q-value of an action $a^{'}$ from a next state $S^{'}$ ($Q_{target}(s', a')$) utilized to estimate the action's future value (see Figure 7). The reference Q-value for an arbitrary state s and an action a (i.e., $Q_{reference}(s, a)$) is calculated as follow: $Q_{reference}(s, a) = r(s, a) + \gamma . max_{a'}\, Q_{target}(s', a')$, where $r(s, a)$ is the immediate reward, and $max_{a'}\, Q_{target}(s', a')$ is the highest Q-value computed by the target network for the next state s', and any action a from the set of legal actions. γ is also a discount factor set to 0.99.

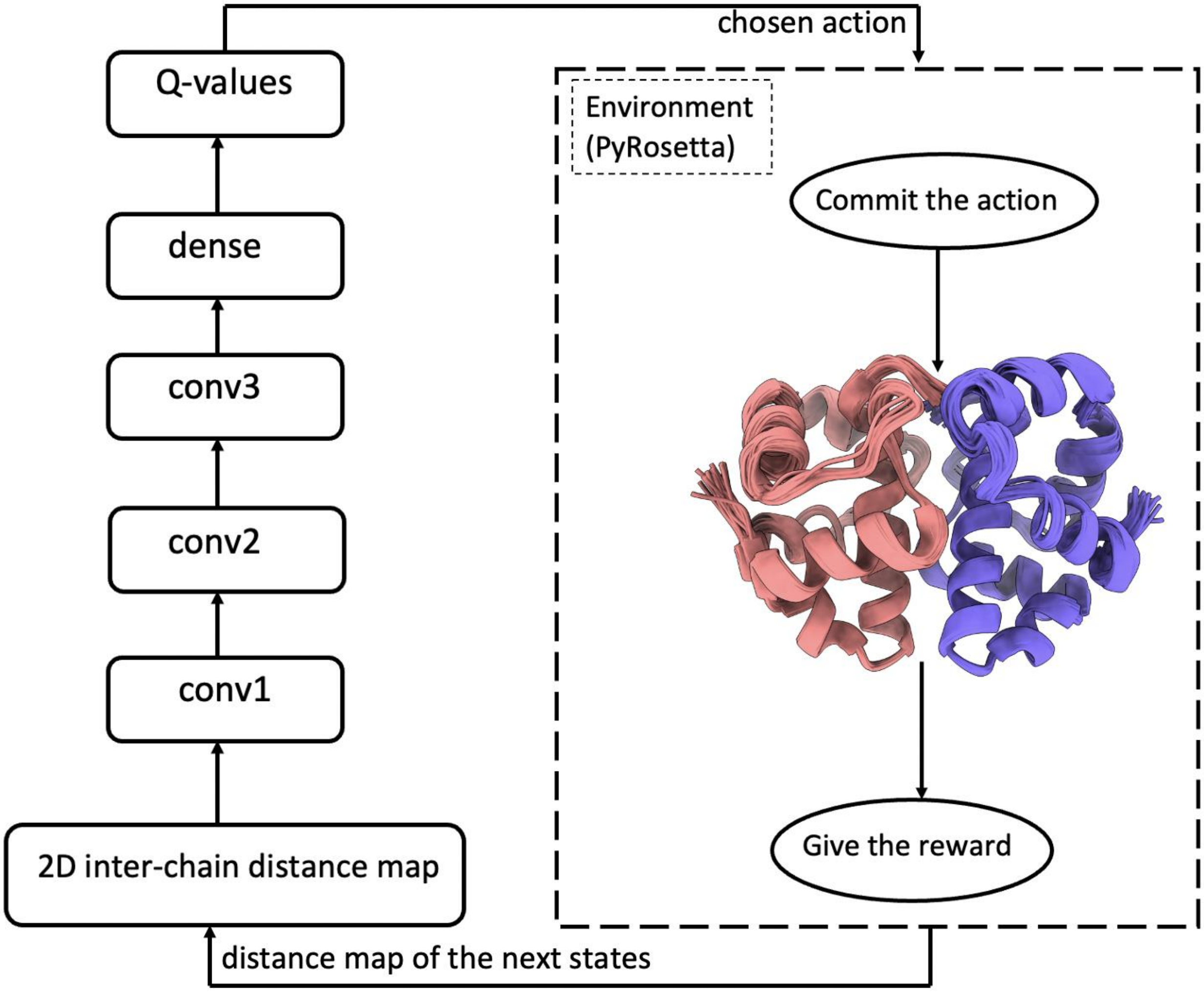

Figure 5. The CNN network to predict the Q-values for every action from the inter-protein distance map of an input state S. The first convolutional layer contains 16 3×3 filters using a stride of 2. The second convolutional layer has 32 filters of size 5×5 with a stride of 2. The third convolutional layer also consists of 64 filters of size 3×3 using a stride of 2. All convolutional layers use a rectified linear unit (ReLU) activation function. The output of the third convolutional layer is flattened and passed to a dense layer (fully connected layer) with a linear activation function to output the Q-values of the individual actions.

The loss function to optimize the deep CNN network is simply the expected squared error defined as $(\frac{1}{N}\sum_{S_i} \quad (Q(s,a) - Q_{reference}(s_i,a_i))^2)$, where N represents the number of states.

The model is trained using stochastic gradient descent (minibatch sizes of 16, 32, and 64). During the training process, ε of the ε-policy decays until it reaches 0.1, and then keeps fixed thereafter.

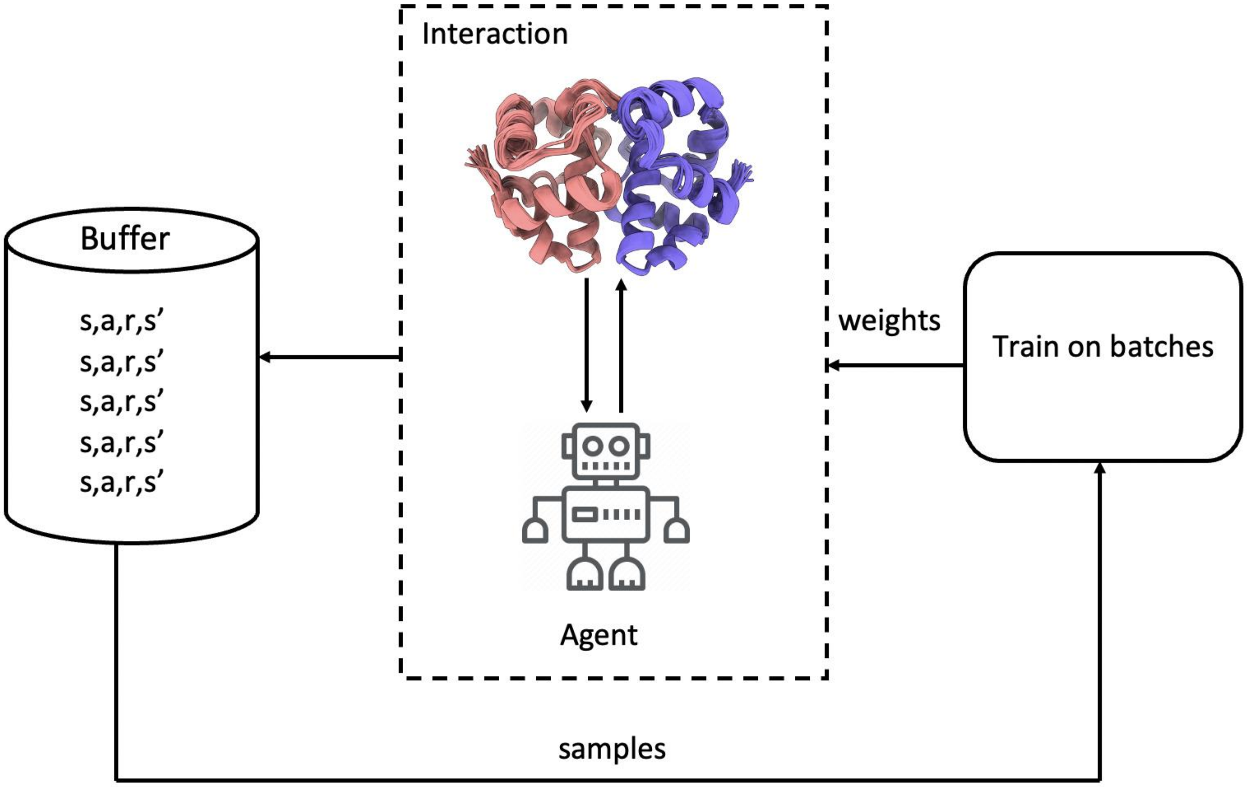

Figure 6. Training procedure for a protein complex using the experience replay technique. Through continuous interaction with the environment, the agent collects a sequence of past experiences (states, actions, rewards, and next states), which are kept in a buffer. When the buffer gets full, some of the old experiences should get dropped to release its space for new experiences. To update the Q-function, the network is trained on K experiences sampled from the replay buffer. The process continues until the agent meets a final state.

We investigated two basically different strategies to examine the quality of a state (S) to compute the immediate reward: (1) the root mean square deviation (RMSD) between the current 3D conformation of S and the native structure (S^*), and (2) an energy function to measure the agreement between the contact map derived from the current 3D structure of S and a predicted contact map passed to the system at the beginning of episodes. Since native structures are often not avaiable in real-world applications, our final set of experiments have been carried out using the latter strategy (contact energy). For the first strategy, the immediate reward (r (s, a)) is calculated according to the formula $RMSD_S$ - $RMSD_{S'}$, where $RMSD_S$ represents the RMSD between the 3D structure of S and the reference structure S^*, and $RMSD_S$ defines the RMSD between the 3D structure of the next state $S^{'}$ and the native structure S^*. If taking action a improves the quality of the state (i.e., $\mathrm{RMSD}_{S'} < \mathrm{RMSD}_S$), then the agent receives a positive reward, otherwise, it is penalized with a negative reward. Figure 8 illustrates the accumulated reward and RMSD of the structure at each episode for a dimer. It is shown that the accumulated reward converges to a positive value, and the agent generates a high-quality structure (RMSD of 0.29 Å).

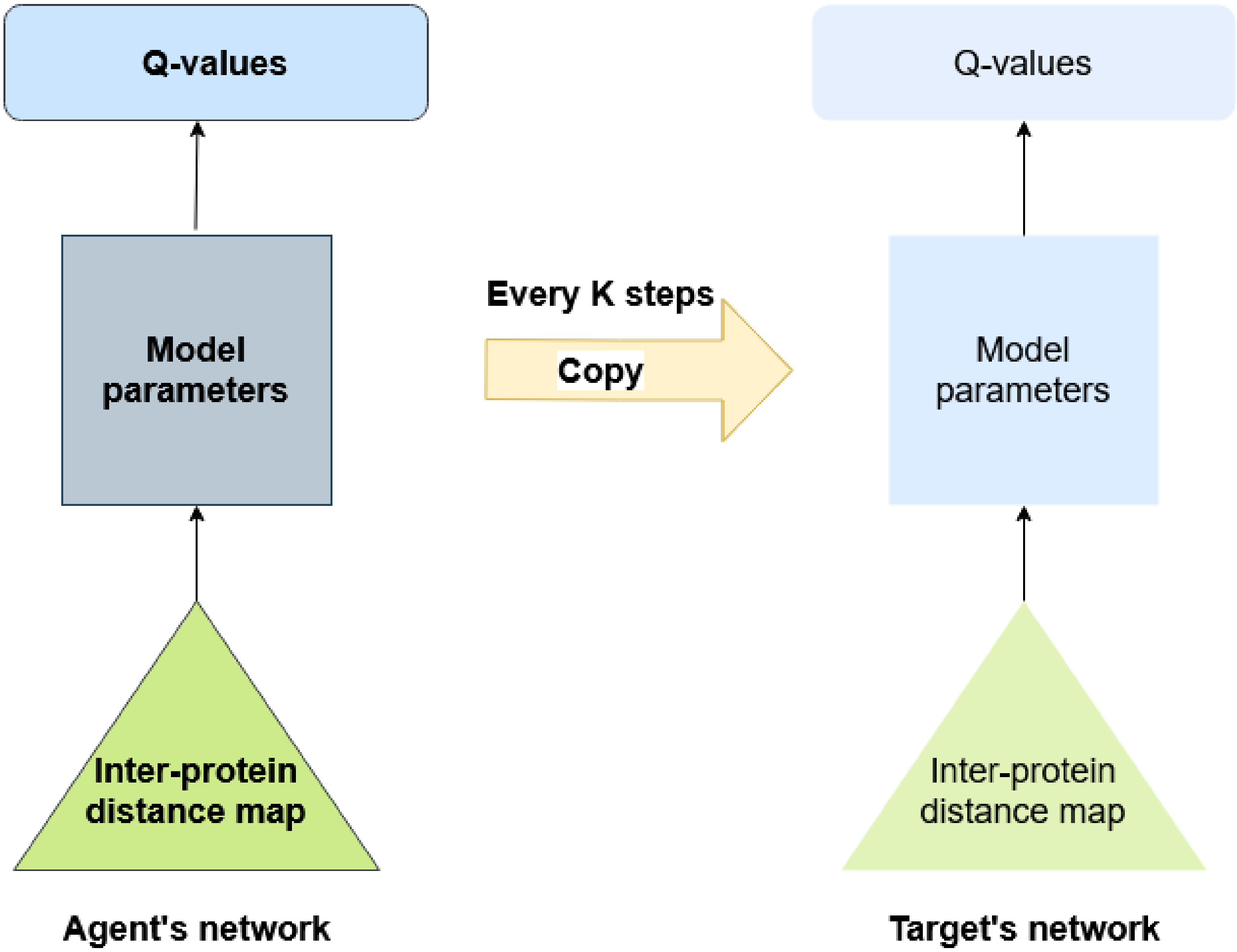

Figure 7. The agent's deep CNN network (Q) copies its weights to the target network ($Q_{reference}$) every K steps. The Q-values predicted by the target network are used as reference Q-values to train Q.

For the second strategy, the immediate reward is defined as the difference between the contact energies of S and S'. The contact energy function is the same as the energy function used by GD and MC. Here, the main goal of the agent is to pick actions, which maximize the satisfaction of the contact restraints within the 3D conformation of a given protein dimer. Figure 9 demonstrates the changes in the accumulated reward and the RMSD of the generated structure across the learning episodes when the contact energy is utilized to compute immediate rewards. As can be seen from Figure 9, the reward converged after 200 episodes. The self-training finished by reconstructing a structure having a RMSD of 0.94 Å. The experiment proves that high-quality models can be reconstructed by the self-learning algorithm whose primary objective is to maximize the rewards based on a provided contact map. The deep CNN was trained for 100K steps, and the target network ‘s weights got updated every 500 steps.

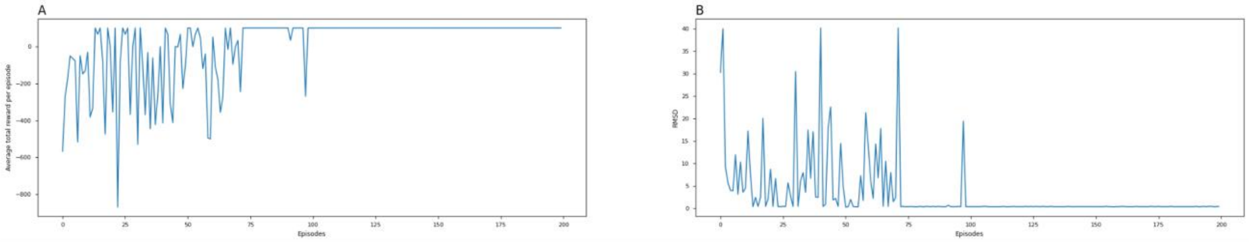

Figure 8. Deep network results for 1A2D dimer using the structural distance between the current conformation of the ligand and its true conformation measured by RMSD as the reward function. (A) The total reward at each episode averaged over three self-learning runs (games). The total reward converges after 100 episodes. (B) The RMSD of the structure at each episode. RMSD converges to a low value after about 100 episodes.

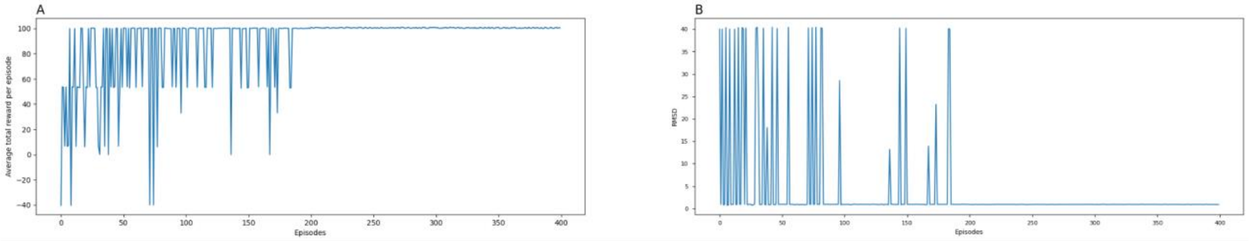

Figure 9. Deep network results for 1A2D dimer using contact energy as the reward function. (A) The accumulated reward per episode. (B) the RMSD of the generated model at the end of each episode.

Gradient descent method to build protein complexes (multimers) from distance constraints:

We create a new version of our GD method to directly take distance restraints of multiple chains instead of two chains as input to reconstruct the 3D structure of multimers. To improve the effectiveness and robustness of the optimization, we iteratively add distance restraints for optimization in a stochastic (randomly selecting a group of distance constraints) way. The objective of this gradient descent method is to minimize the sum of square distances between given input distances and the distances in the 3D model. The 3D model is initialized as a random model;

chains are randomly rotated and translated. The x, y, z coordinates of C_α or C_β are adjusted based on the partial derivatives of the objective function with respect to each coordinate.

When a small portion of constraints are randomly selected each time, our method is a kind of stochastic gradient descent optimization, mainly utilized for optimizing deep neural networks. In contrast to utilizing all the constraints at the same time that often gets stuck in bad local minimums, using a small batch of restraints helps the optimization method to escape bad local minimums and quickly converge to good local minimums that may have similar performance as global minimum. The optimization process continues until the RMSD between the models generated in the current and previous epochs is less than 0.1 Å. Figures 10-13 show high-quality models reconstructed by our method.

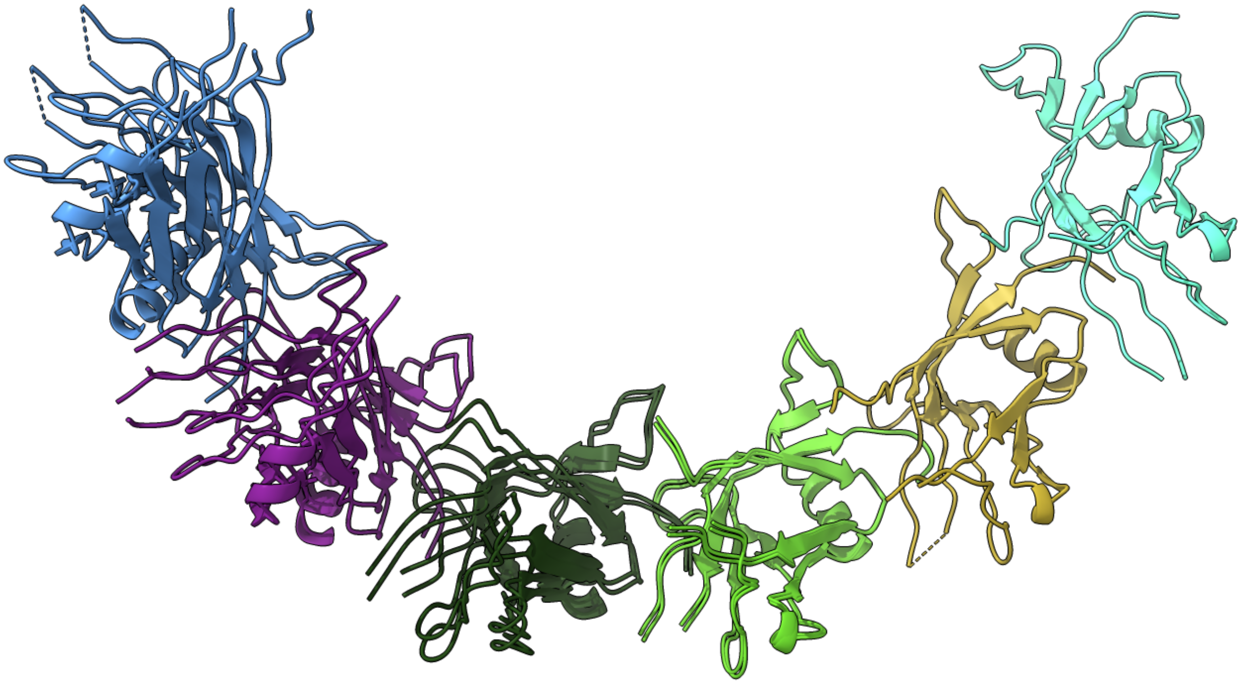

Figure 10. Superposition of the native structure of H1060 and the model predicted by GD for multimers. The RMSD between the two structures is 1.23 Å.

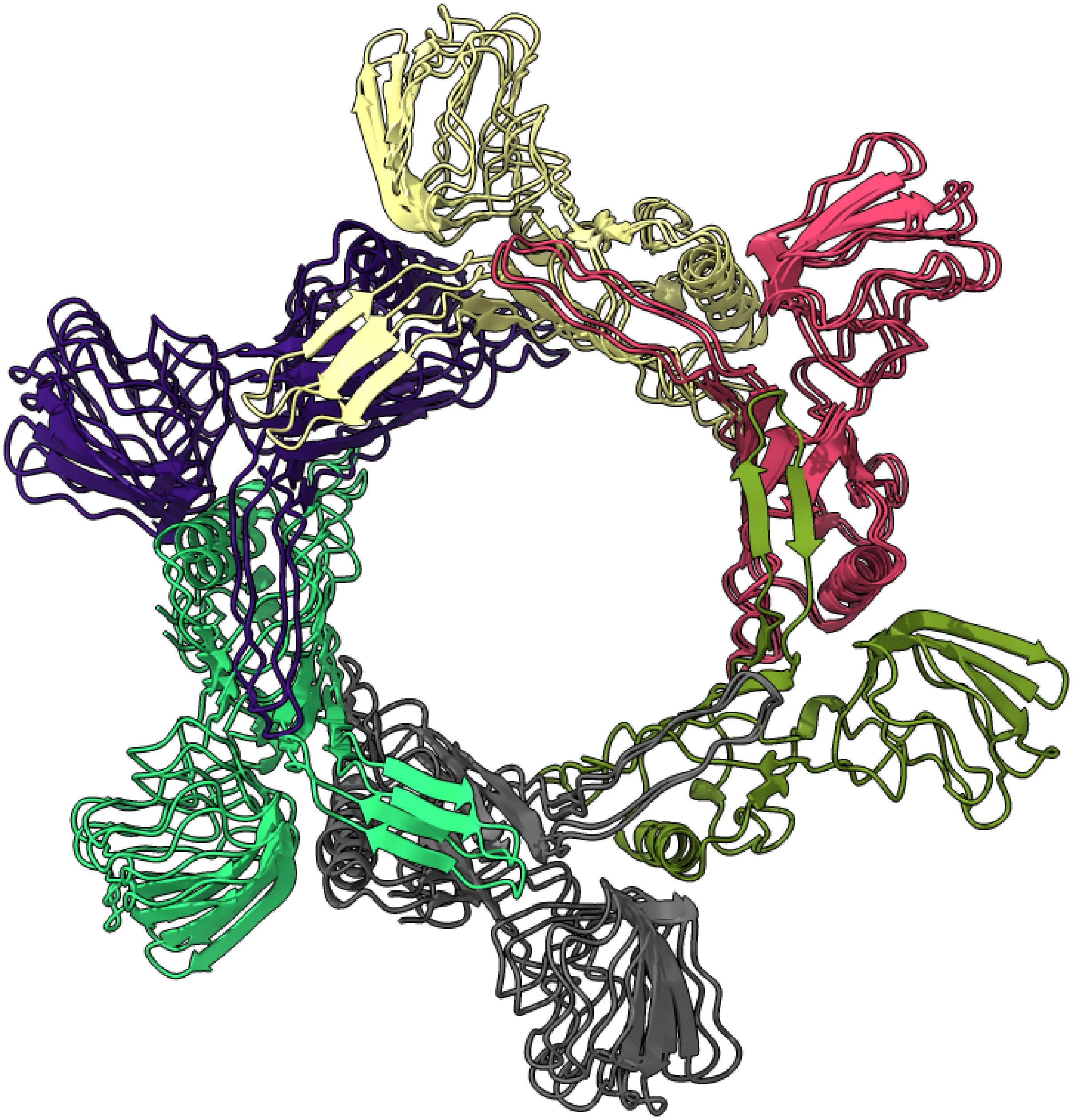

Figure 11. Superposition of the true structure of H1066 and the model reconstructed by GD for multimers. The RMSD between the two structures is 1.18 Å.

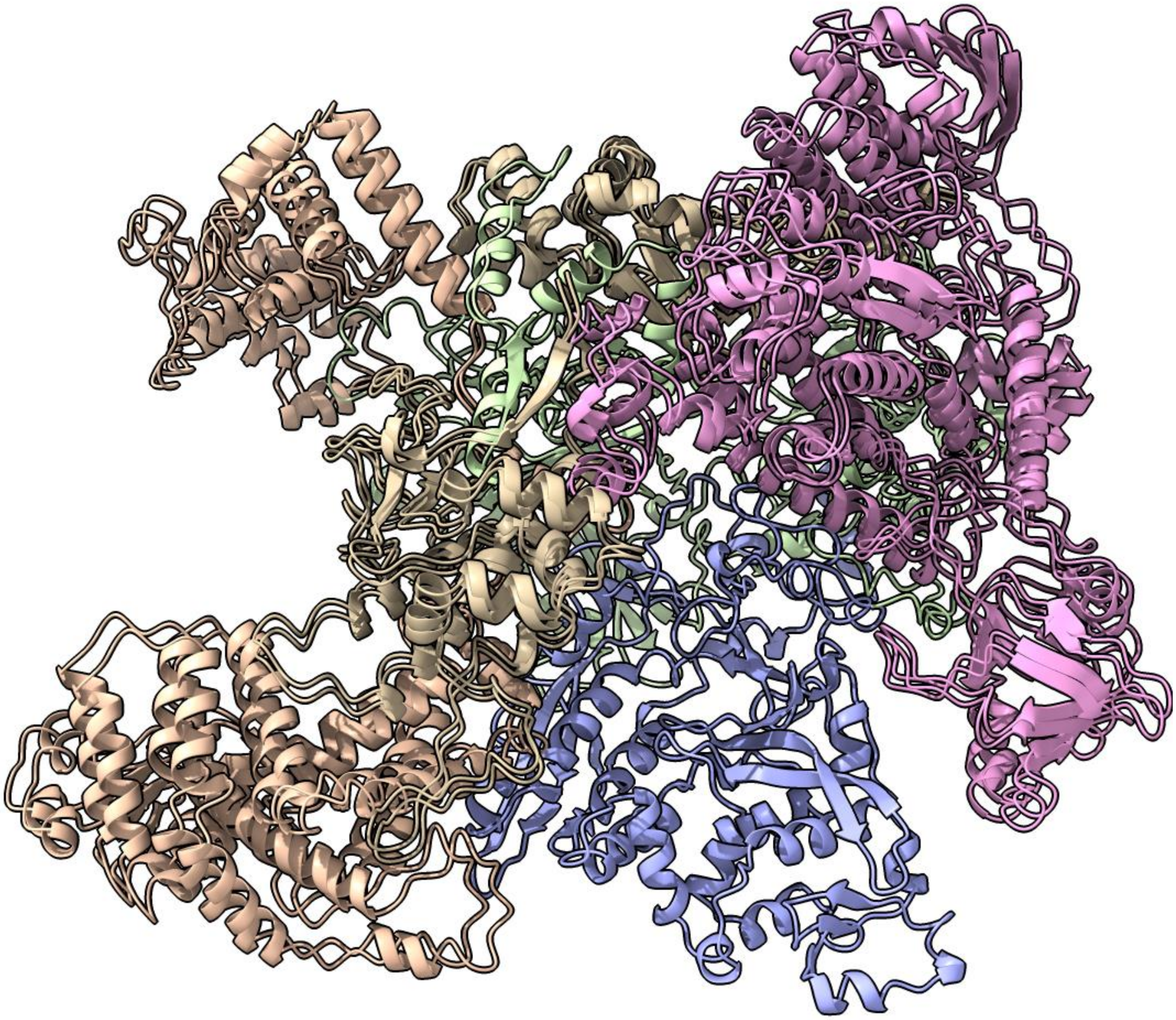

Figure 12. Superposition of the true structure of H1097 and the model reconstructed by GD for multimers. The RMSD between the two structures is 1.29 Å.

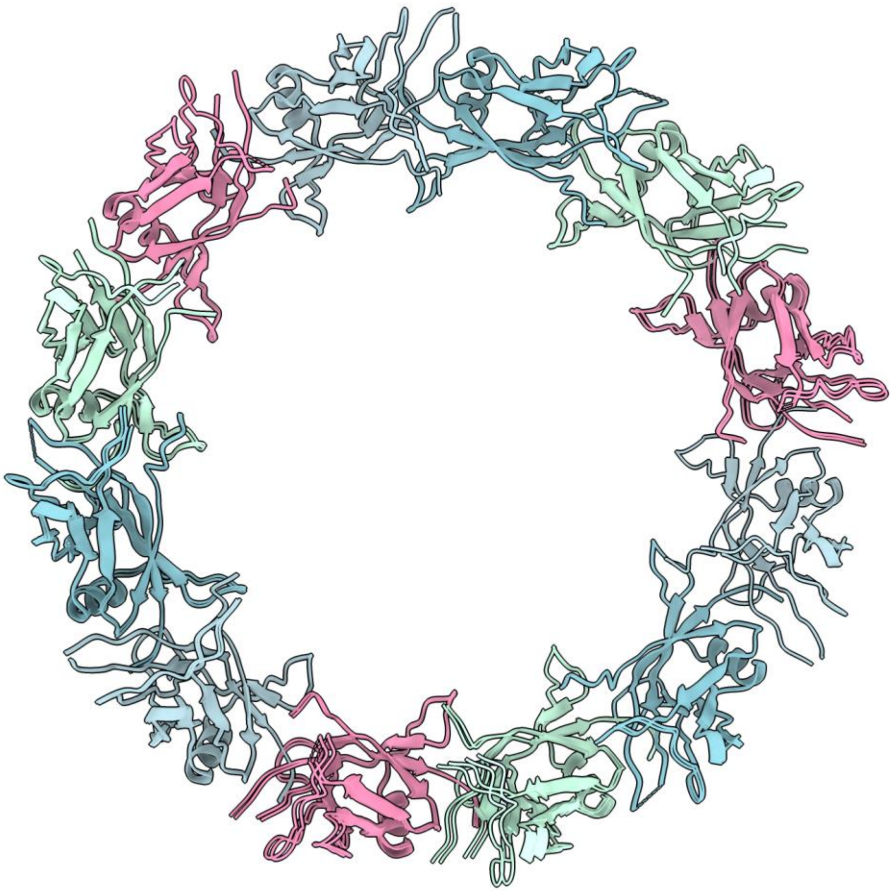

Figure 13. Superposition of the true structure of H1060v4 and the model reconstructed by GD for multimers. The RMSD between the two structures is 0.32 Å.

Results and Discussions of Gradient Descent Method

Inter-protein contacts and protein complex datasets

Two residues are said to be in contact if the distance between their C_β atoms (C_α for Glycine) is less than or equal to 6Å [37, 38]. Native inter-chain contacts are extracted from known quaternary structures.

For these experiments, we use several datasets of homodimers and heterodimers using both true and predicted inter-protein contacts. The first dataset has 44 homodimers (Homo44), randomly selected from Homo_Std [37]. The number of inter-chain contacts for dimers in Homo44 is between 38 to 622. The second dataset, called Homo115, consists of 115 homodimers with at least 21 predicted inter-chain contacts having a probability of greater or equal to 0.5. ResCon [39] is used to predict the inter-chain contacts for the dimers in Homo115. Based on the size of the interaction interface, Homo115 is grouped into three subsets, named Set A, Set B, and Set C. Set A consists of 40 proteins with 14 to 68 true inter-chain contacts (small interaction interface). Set B has 37 dimers with 69 to 129 true inter-chain contacts (medium interaction interface). Set C contains 38 dimers having 131 to 280 true inter-chain contacts (large interaction interface). The third dataset, called Hetero73 [39], has 73 heterodimers with 2 to 255 true inter-chain contacts. The last dataset is Std32, which has 32 heterodimers [39].

Generating protein dimers using true (native) contacts

Firstly, we used GD, MC, and CNS to reconstruct the quaternary structures of protein dimers from native contacts for 44 homodimers (Homo44). To evaluate the quality of the generated models, we use five metrics including root-mean-square deviation (RMSD), TM-score, the percentage of native contacts preserved in predicted models (f_nat), interface RMSD (I_RMSD), and Ligand RMSD (L_RMSD).

Supplementary Table 1 (Table S1) represents the detailed results including RMSD, TM-score, f_nat, I_RMSD, and L_RMSD of the GD method for Homo44. As can be seen from Table S1, using true inter-protein contacts, GD reconstructs high-quality models for all the protein complexes (the TM-score of the predicted models all fall in the ranges of 0.936 to 0.999). Table 1 summarizes the mean values of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD for GD, MC, and CNS. GD achieves the best TM-score, RMSD, f_nat, I_RMSD, and L_RMSD. The average value of RMSD for GD is 0.63Å, lower than those of MC (0.76Å) and CNS (1.16Å). GD achieved an average TM-score of 0.99, 0.01 and 0.08 higher than MC and CNS, respectively. While GD preserves 92.19% of true contacts, 91.39% and 82.49% of native contacts exist in the predicted models by MC, and CNS, successively. Moreover, GD has an average I_RMSD of 0.77Å, lower than 1.35Å of MC and 12.46Å of CNS. Also, GD improved L_RMSD by 0.32 Å and 9.8 Å over the MC and CNS methods. The improvement of RMSD, I_RMSD, and L_RMSD over MC and CNS is more significant, which means GD pays more attention to the accurate reconstruction of atomic coordinates of the generated structure. Figure 14 shows high quality models generated by the three methods (GD, MC, and CNS) for a homodimer from Homo44 using true contacts.

Table 1. Mean and standard deviation (std) of RMSD, TM-score, f_nat, I_RMSD, and L_RMSD of GD, MC, and CNS for Homo44 using true contacts as restraints.

Evaluation Metric	**GD**	**MC**	**CNS**

RMSD (mean, std)	**$0.63^{+-0.3788}$**	$0.76^{+-0.361}$	$1.16^{+-1.0043}$
TM-score (mean, std)	**$0.99^{+-0.0132}$**	$0.98^{+-0.014}$	$0.91^{+-0.0102}$
f_nat (mean, std)	**$92.19^{+-8.64}$**	$91.39^{+-9.08}$	$82.49^{+-22.02}$
I_RMSD (mean, std)	**$0.77^{+-1.05}$**	$1.35^{+-3.98}$	$12.46^{+-8.46}$
L_RMSD (mean, std)	**$1.38^{+-0.8}$**	$1.7^{+-0.9}$	$11.18^{+-14.51}$

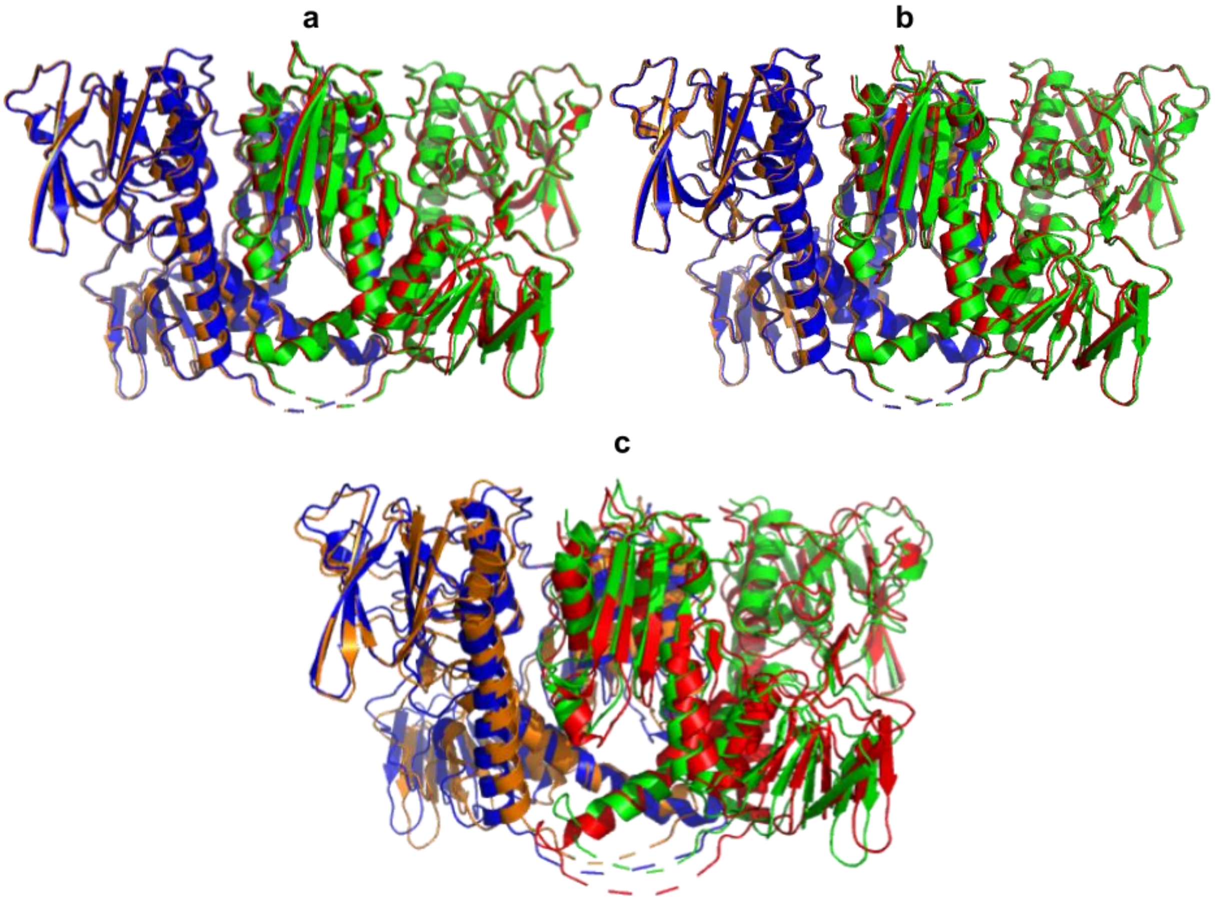

Figure 14. The superposition of the native structure of a homodimer (1XDI), and the models built by GD, MC, and CNS. Green and orange show the two chains of the true structure, whereas blue and red represent the two chains of the predicted models. (a) the model reconstructed by GD using true inter-protein contacts. The detailed results of the model in (a) are as follow: TM-score = 0.99, RMSD = 0.56Å, f_nat = 94.52%, I_RMSD = 0.24Å, and L_RMSD = 0.74Å. (b) the model built by MC utilizing true inter-chain contacts. The detailed results of the generated model in (b) are the following: TM-score = 0.99, RMSD = 0.61Å,

f_nat = 93.15%, I_RMSD = 0.45Å, and L_RMSD = 1.29Å. (c) The model reconstructed by CNS using true inter-chain contacts. The detailed results for the model in (c) are TM-score = 0.88, RMSD = 2.25Å, f_nat = 74.79%, I_RMSD = 1.49Å, and L_RMSD = 5.18Å.

We further evaluated the performance of the three methods on a dataset of 77 heterodimers (Hetero77) when true inter-protein contacts are provided as constraints. Table 2 compares GD, MC, and CNS in terms of the metrics mentioned above. Detailed per complex results are also shown in Table S2. As shown in Table 2, GD outperforms the two other methods in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD (GD achieves higher TM-score and f_nat, and lower RMSD, I_RMSD, and L_RMSD than MC and CNS). Overall, given inter-chain contacts as input, GD is able to build high quality models for heterodimers as the mean TM-score and RMSD of the reconstructed models are 0.92, and 1.23Å, respectively (see Table 2). However, compared to the models reconstructed for homodimers (see Table 1), the predicted models for heterodimers are of lower quality. One possible reason for this difference in the performance is that heterodimers often have lower inter-protein contact density (i.e., number of inter-protein contacts over the whole length of a protein complex), and consequently, fewer contact restraints provided for the optimization process.

Table 2. Mean and standard deviation (std) of RMSD, TM-score, f_nat, I_RMSD, and L_RMSD results of GD, MC, and CNS for 77 heterodimers when true inter-chain contacts are given as input.

Evaluation Metric	**GD**	**MC**	**CNS**
RMSD (mean, std)	**$1.23^{+-1.91}$**	$4.76^{+-8.01}$	$7.7^{+-12.99}$
TM-score (mean, std)	**$0.92^{+-0.12}$**	$0.85^{+-0.16}$	$0.79^{+-0.23}$
F_nat (mean, std)	**$90.31^{+-16.77}$**	$82.59^{+-26.68}$	84.43^{+-23}
I_RMSD (mean, std)	**$0.72^{+-1.02}$**	$1.58^{+-1.7}$	$1.65^{+-4.51}$
L_RMSD (mean, std)	**$3.75^{+-6.15}$**	$7.78^{+-11.8}$	$9.21^{+-14.05}$

Table 3. Mean RMSD, TM-score, f_nat, I_RMSD, and L_RMSD of the three methods for Std32 containing 32 heterodimers.

Evaluation Metric	GD	MC	CNS
TM-score	**0.96**	0.95	0.82
RMSD	**1.95**	2.9	10.04
f_nat	**92.78**	92.43	69.13
I_RMSD	**1.64**	1.99	3.71
L_RMSD	**4.65**	7.16	-14.99

Furthermore, we evaluated the three methods on another dataset, named Std32, which has 32 heterodimers. Detailed results of GD for this dataset using true inter-chain contacts are reported in Table S3. Mean values of RMSD, TM-score, f_nat, I_RMSD, and L_RMSD are also reported in Table 3. These results are consistent with the previous results as GD has the best performance among the three methods, and MC reconstructs higher accurate models than CNS.

The effects of initial start, and contact density on the quality of the models built by GD using true inter-chain contacts as input

Initial conformation and inter-protein contact density impact the quality of the final reconstructed structure.

Figure 15 demonstrates how the TM-score and RMSD of the generated structures change with respect to the initial conformations for a protein dimer. As can be seen in Figure 15, TM-scores of the generated structures fall in a range of 0.55 (a very poor score) to 1 (an almost perfect score), indicating that starting from a reasonable initial conformation, GD has the ability to converge to a local minimum with almost the same performance as the global minimum and, as a result, generate a high-quality model. By contrast, given a poor initial structure, GD might get stuck in a bad local minimum and return a low-quality model. It is therefore useful to repeat the optimization process with different initial starts. In fact, our experiments on Homo44 and Hetero77 show that repeating the optimization process with 20 different initial starts helps the GD method to reconstructs high quality models having a TM-score of 0.99 and RMSD of less than 1Å when true inter-chain contacts are given as input (see Table S1 and Table S2).

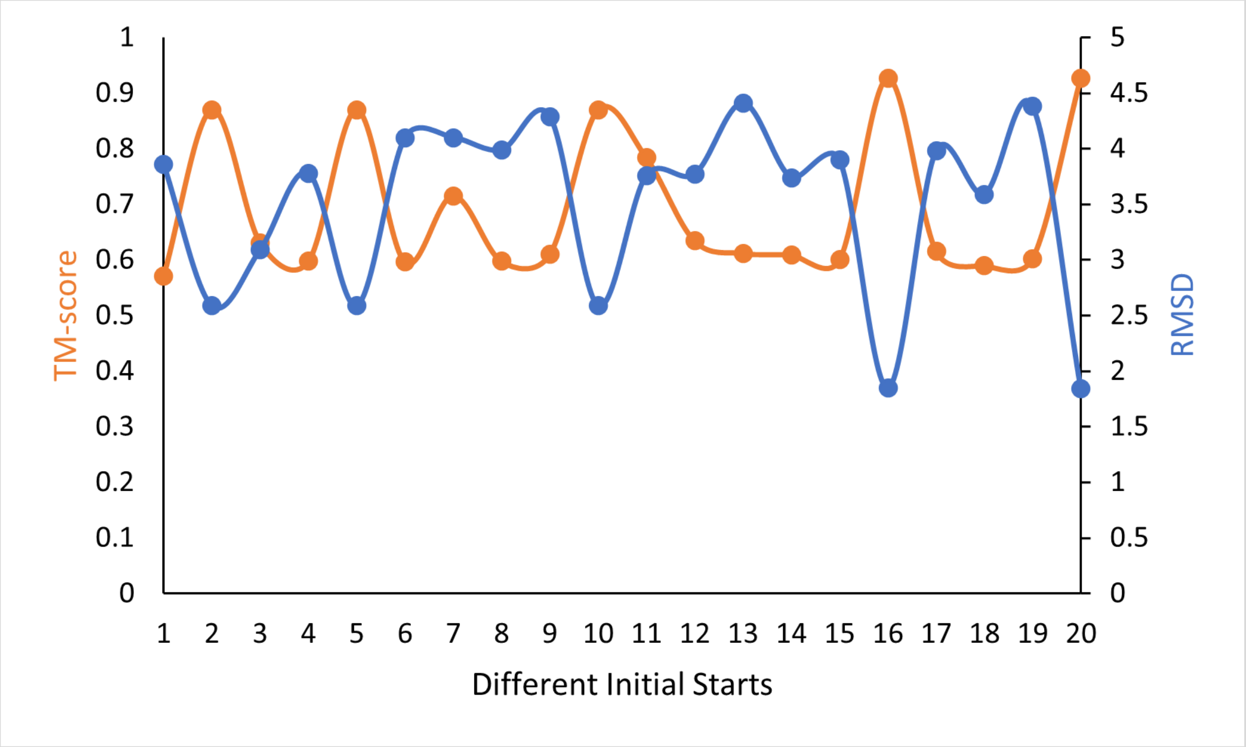

Figure 15. TM-score and RMSD of the GD method for several models starting from different initial conformations for a homodimer (PDB code: 1Z3A). X-axis shows different initial starts for the optimization process. Y-axis denotes the quality of the reconstructed models in terms of TM-score and RMSD.

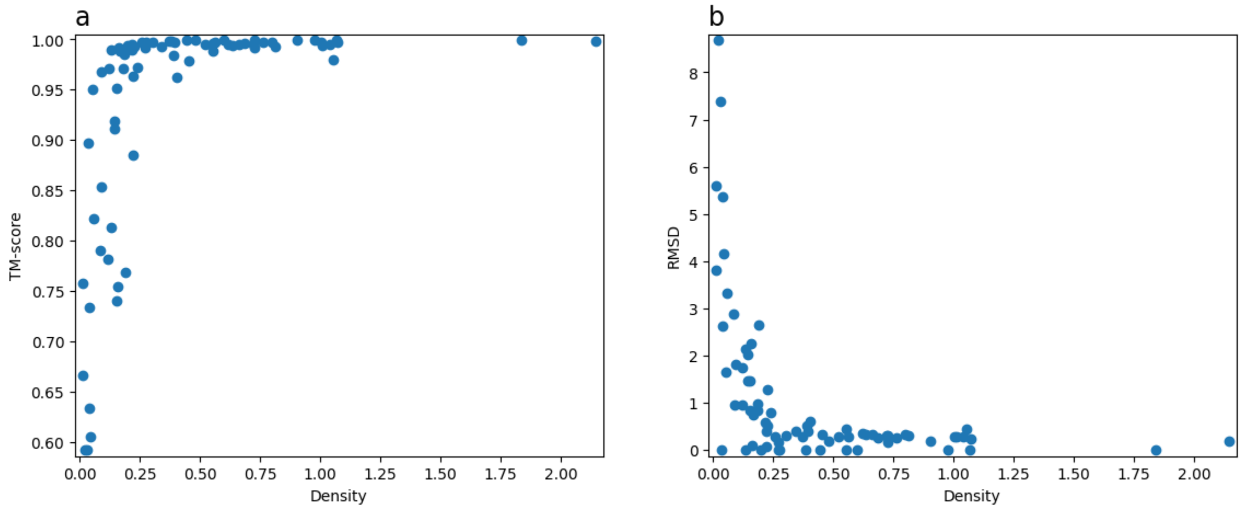

Figure 16. Inter-protein contact density against the TM-score and RMSD of the generated models for Hetero77.

Additionally, the quality of the final structural model is highly affected by the density of the provided inter-chain contact map. Figure 16 demonstrates the changes in the TM-score and RMSD of the generated models for different contact densities. Based on Figure 16, for protein dimers with a contact density of 0.25 or higher, GD reconstructs high quality models with TM-score of about 1 and RMSD of less than 1Å. On the other hand, when the contact density is less than 0.25, GD fails to build good quality structures for some of the protein dimers in Hetero77. In general, the higher the contact density of a dimer, the higher the quality of the generated models (higher TM-score and lower RMSD).

Building protein homodimers from predicted inter-chain contacts

We also evaluated the performance of GD, MC, and CNS on three sets of protein complexes, named Set A, Set B, and Set C, when predicted inter-protein contacts are provided as restraints. Set A has 40 protein complexes having a small interaction interface. Set B contains 37 complexes having medium interaction interface. Set C has 38 protein dimers with extensive interaction interface. Tables S4, S5, and S6 report the results of GD in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD, and also the quality of the provided, predicted contacts in terms of precision and recall. Precision and recall are calculated as $\frac{\#\, correct\ contacts\ with\ probability >= p}{\#contacts\ with\ probability >= p}$ and $\frac{\#\, correct\ contacts\ with\ probability >= p}{\#true\ contacts}$, respectively, where p is the cut-off probability, set to 0.5.

Table 4. Mean values of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of GD, MC, and CNS for Set A using predicted inter-chain contacts.

Evaluation Metrics	**GD**	**MC**	**CNS**
TM-Score	**0.68**	0.66	0.58
RMSD	**10.81**	11	17.48
f_nat	**22.47**	18.38	14.67
I_RMSD	**9.93**	10.03	12.37
L_RMSD	**25.46**	27.81	-30.35

Table 5. Mean values of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of GD, MC, and CNS for Set B using predicted inter-chain contacts.

Evaluation Metrics	**GD**	**MC**	**CNS**
TM-score	**0.8**	0.77	0.64
RMSD	**6.78**	8.3	12.89
f_nat	**32.18**	28.66	22.19
I_RMSD	**6**	7.6	13.3
L_RMSD	**14.87**	18.46	-20.69

Table 6. Mean values of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of GD, MC, and CNS for Set C using predicted inter-chain contacts.

Evaluation Metrics	**GD**	**MC**	**CNS**

TM-score	**0.81**	0.80	0.76
RMSD	**6.26**	6.77	9.5
f_nat	37.43	35.07	**42.3**
I_RMSD	**5.01**	5.46	7.41
L_RMSD	**12.73**	13.96	-16.3

The averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of the three methods are shown in Tables 4, 5, and 6. These results, which are consistent with previous results, indicate that GD performs better than MC, and CNS in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD. Furthermore, as the interaction interface extends, the quality of the reconstructed models also increases (i.e., quality of the models for Set C > quality of the models for Set B > quality of the models for Set A), indicating that it is much easier to build protein complexes having extensive interaction interface. As shown in Tables 4, 5, and 6, the averaged TM-score of GD for the models built for Set A, Set B, and Set C are 0.68, 0.80, and 0.81, respectively, superior to the other methods. The average TM-score of the GD method for all the protein complexes in Set A, Set B, and Set C (in total 115 protein complexes) is 0.76, showing that GD can reconstruct reasonable models for most of the given protein dimers. In addition, based on Tables S4, S5, and S6, GD was able to reconstruct high quality models (TM-score > 0.9) for about 46% of protein dimers. Figure 17 shows a high-quality model built by GD for a protein complex (PDB code: 1C6X) using inter-chain contacts with precision of 40.24% and recall of 49.28%. The model in Figure 17 has a TM-score of 0.99 and f_nat of 84.61%.

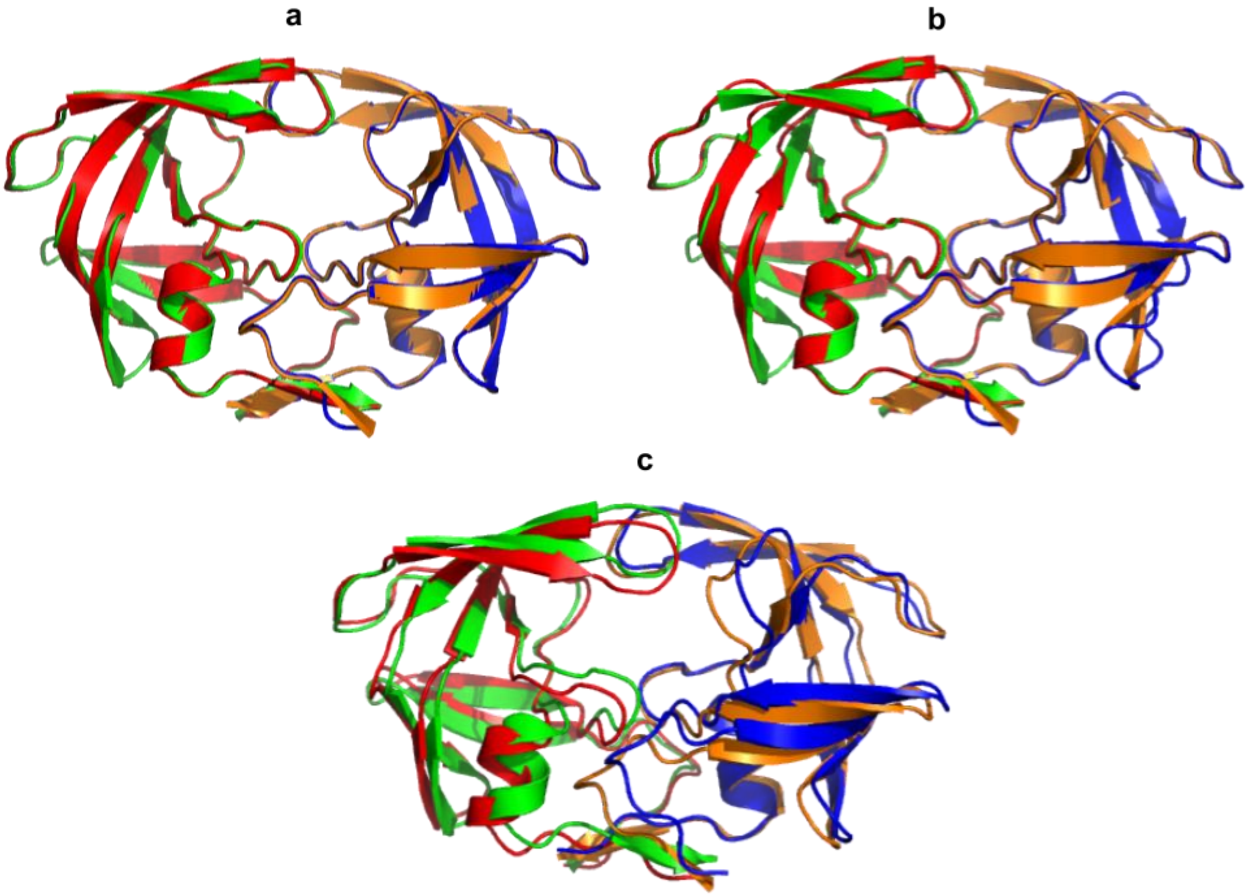

Figure 17. The superposition of the true structure and the high-quality models built by GD, MC, and CNS. The two chains of the true structure are represented by green and orange, whereas the two chains of the reconstructed model are denoted by blue and red. (a) The high quality model built by GD with TM-score=0.99, RMSD=0.4Å, f_nat=84.61%, I_RMSD=0.4Å, and L_RMSD=0.91Å. (b) The model built by MC with TM-score=0.98, RMSD=0.6Å, f_nat=78.84%, I_RMSD=0.6Å, and L_RMSD=1.6Å. (c) The model built by CNS with TM-score=0.86, RMSD=2.02Å, f_nat=41.6%, I_RMSD=2.14Å, and L_RMSD=5.68Å.

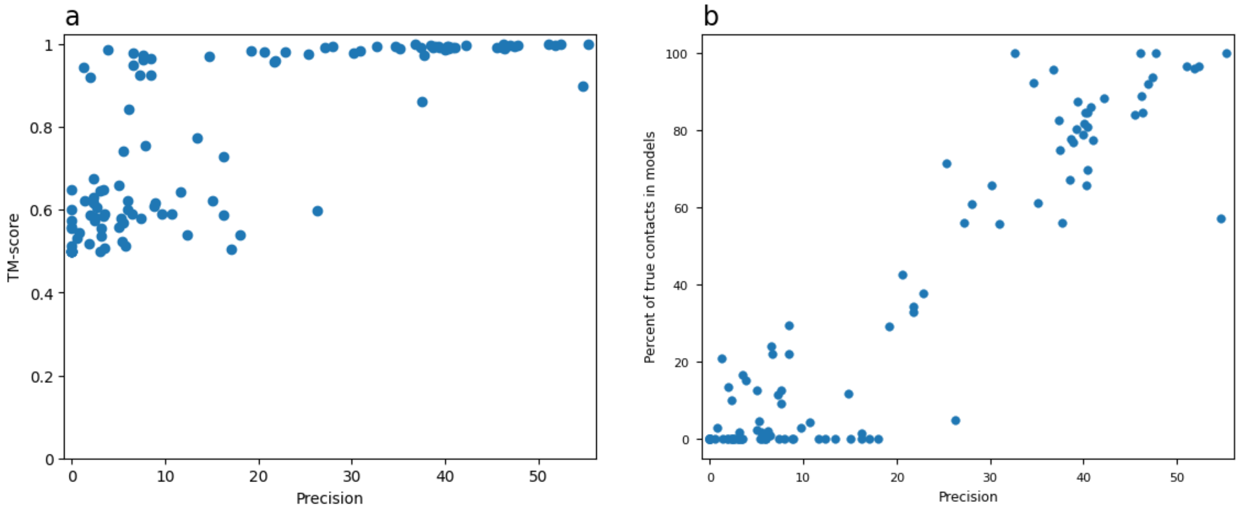

Figure 18. (a) TM-score of the models reconstructed by GD vs the precision of the inter-chain contacts for 115 complexes in Set A, Set B and Set C. (b) f_nat of the reconstructed models against the precision of the inter-chain contacts for 115 dimers in Sets A, B, and C.

We also investigated two factors that affect the quality of the models built by GD: precision and recall of the predicted inter-chain contacts. Figure 18a illustrates how the TM-score of the reconstructed models change with respect to the precision of the predicted inter-chain contacts. The correlation between TM-score values and contact precisions is 0.78, indicating that the higher the precision the more accurate the models. According to Figure 18a, if the precision is higher than 20%, then the majority of the models have TM-scores equal or more than 0.8. When the precision reaches to 40%, all the generated models are of high quality (TM-score > 0.8). Therefore, it seems that moderately accurate inter-chain contacts are quite sufficient to build high quality models for dimers. Besides, it indicates that the GD method is not sensitive to noise present in predicted contacts.

Figure 18b also demonstrates that f_nat values of the reconstructed models are highly dependent on the precision of the predicted inter-chain contacts as the correlation between these two is 0.94. If the precision is higher than 40%, then f_nat values fall into 50% to 100%.

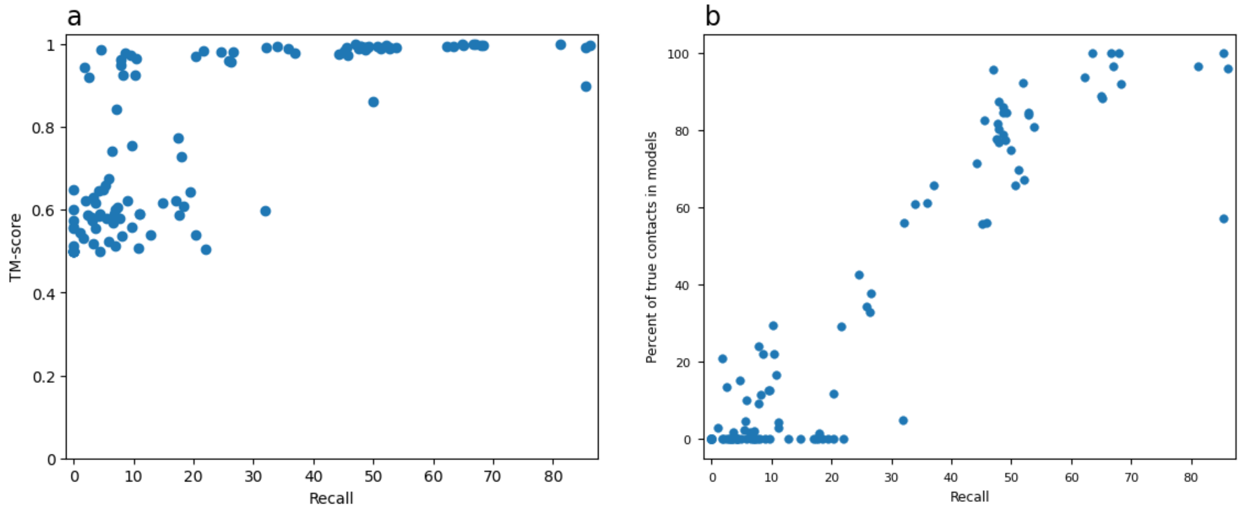

Figure 19. TM-score and f_nat values of the models reconstructed by GD for 115 complexes in Sets A, B, and C against the recall of the given predicted inter-chain contacts.

Also, TM-score and f_nat values of the models built by GD are highly affected by the recall of predicted inter-chain contacts (see Figure 19). Based on Figure 19, TM-score and f_nat are highly correlated with the recall of the predicted contacts (correlation between TM-score and recall is 0.78, and correlation between f_nat and recall is 0.93). It is evident from Figure 19a, similar to the previous findings, that a recall of 40% is enough to reconstruct high quality models having a TM-score of more than 0.8 and f_nat of larger than 50%.

Moreover, we investigated the relationship between the cut-off probability (used to exclude unwanted inter-chain contacts) and the quality of the generated models for 115 complexes in Set A, Set B, and Set C. Figure 20 reports the TM-score and RMSDs for cut-off probabilities from 0.3 to 0.9 with a step size of 0.1. Based on Figure 20, the highest TM-score (0.77) and lowest RMSD (about 7Å) was achieved by using a cut-off probability of 0.5. However, it must be noted that the cut-off probability is a data-dependent parameter, hence, it seems there is no good way to pick the best value other than trying different cut-off probability and selecting the one that leads to the best performance.

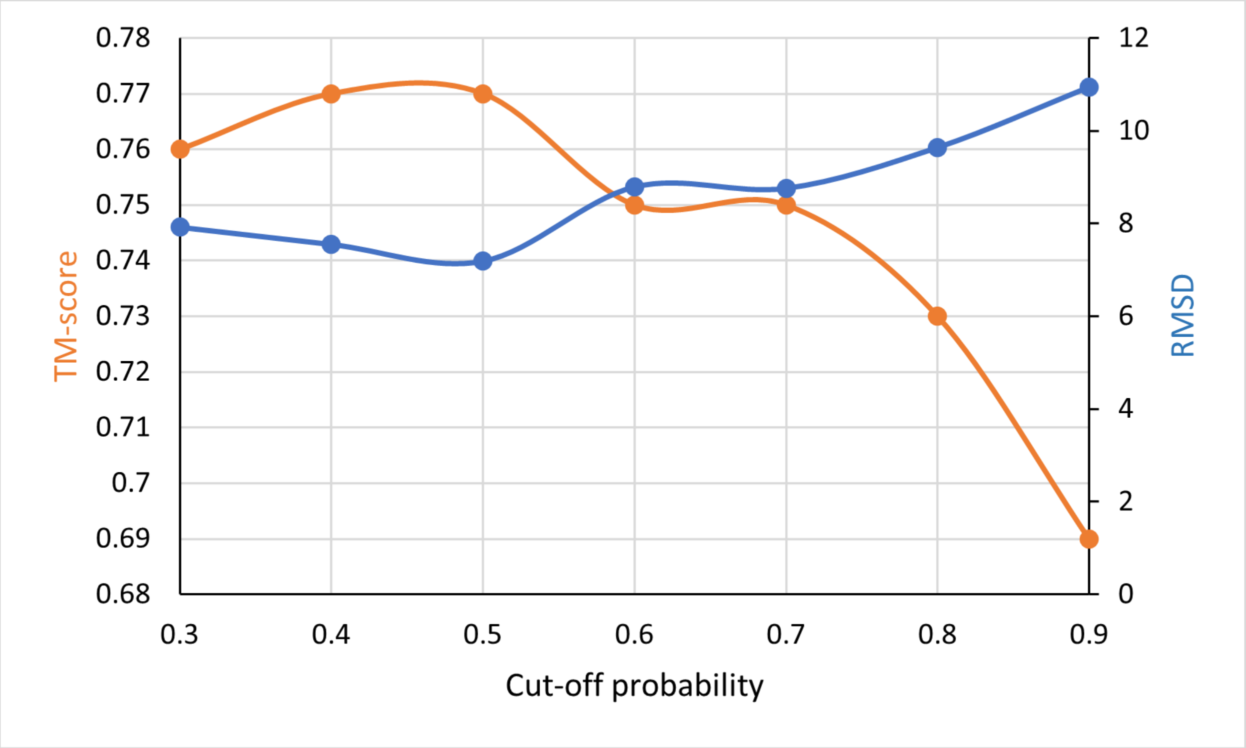

Figure 20. The mean TM-score and RMSD of the model reconstructed by GD for 115 dimers in Sets A, B, and C against the cut-off threshold for discarding unwanted inter-chain contacts.

Results and Discussions of DRLComplex

We evaluate the performance of our reinforcement learning approach and compare its effectiveness with other quaternary structure prediction methods including GD, MC, CNS, and Equidock [40]. We perform experiments on two standard datasets: CASP-CAPRI dataset of 28 homodimers, and Std32 of 32 heterodimers. Specifically, we test two different scenarios, in the first scenario, named optimal scenario, native inter-protein contacts are utilized to guide the modeling process, whereas the second scenario (suboptimal scenario) uses predicted inter-protein contacts to build quaternary structures of protein dimers. As discussed earlier, two residues are said to form a contact if the distance between their heavy atoms is less than or equal to 6Å; however, for simplicity, we consider two residues to be in contact if the distance between their C_β (C_α for Glycine) is within 6Å. Predicted inter-chain contacts are provided by two different methods: DRCon [41] for homodimers, and Glinter [42] for heterodimer. It must be noted that one protein complex from Std32 (PDB code: 1IXRA_1IXRC) is excluded from the experiments as its ligand does not interact with its receptor (there are no contacts between the ligand and the receptor). We also perform a significantly more challenging experiment, called realistic scenario hereafter, by using both predicted monomer structures and predicted inter-chain contacts as the input to the algorithm. AlphaFold2 [43] is utilized to predict monomer structures.

Evaluation Metrics

We compare the performance of the methods in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD. TMalign and Ca-RMSD from Pyrosetta are used to compute TM-score and RMSD, respectively. DockQ is also used to calculate f_nat, I_RMSD, and L_RMSD.

Comparison and evaluation of five methods on 28 homodimers (CASP-CAPRI dataset)

Table 7, Table 8, and Table 9 report a comparative summary of the results of different methods for optimal scenario, suboptimal scenario, and realistic scenario, respectively, on 28 homodimers from CASP-CAPRI dataset. Detailed results including TM-score, RMSD, f_nat, I_RMSD, and L_RMSD for each target are also reported in Tables S7, S8, and S9 in Supplementary materials. Using native inter-chain contacts and true monomer structure as inputs (optimal scenario), DRLComplex outperforms all the methods including GD, MC, and CNS in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD (see Table 7). Based on Table 7, GD also performs better than MC, and CNS. DRLComplex builds high quality models with a f_nat value close to 1 for about 36% of the targets (10 out of 28). The mean f_nat value of the models reconstructed by DRLComplex (shown in Table 7) is 99.05% for CASP-CAPRI dataset, proving that DRLComplex is able to generate models close to the native ones when native inter-protein contacts are provided. The Equidock method does not require any contact information to perform the modeling process, instead it learns how to rotate and translate the ligand with respect to the receptor, thus, we did not consider this method for the optimal scenario.

When predicted inter-protein contacts are provided as input (suboptimal scenario), DRLComplex, GD, and MC give us relatively similar performance as their TM-scores are 0.73, 074, and 0.72, respectively (see Table 8 for other metrics). However, these three methods significantly outperform CNS and Equidock. According to Table 8, DRLComplex achieves an averaged f_nat of 33.21%, 4.89% and 26.7% higher than those of CNS and Equidock, successively.

Moreover, when both predicted inter-chain contacts and monomer structures are used (realistic scenario), DRLComplex achieves superior performance than all the methods, except GD, whose performance is comparable to DRLComplex, even though GD is slightly better than DRLComplex. The averaged f_nat value of the models built by DRLComplex is 27.1%, higher than 16.81% of MC, 13.58% of CNS, and 22.27% of Equidock. In addition, the mean RMSD of the models reconstructed by DRLComplex is 0.6Å, 2.37Å, and 14.41Å lower than MC, CNS, and Equidock. DRLComplex has an averaged TM-score of 0.64, higher than those of MC (0.63), CNS (0.62), and Equidock (0.5).

As shown in Table 7, and 8, using predicted inter-chain contacts, rather than the native contacts extracted from the native monomers, the mean TM-score of the models generated by DRLComplex decreases from 0.989 (almost a perfect score) to 0.73, indicating that the quality of the provided contacts affects the performance of the quaternary structure prediction methods (here DRLComplec, GD, MC, and CNS).

Table 7. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, and CNS using true inter-chain contacts for CASP-CAPRI dataset. Known tertiary structures are used as monomers.

Methods	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	**0.9895**	**0.3753**	**99.05**	**0.2197**	**0.8235**
GD	**0.9895**	**0.3753**	99.03	0.3468	**0.8235**
MC	0.9631	1.2089	78.91	1.1611	2.8897
CNS	0.9234	2.003	73.45	3.9234	4.6841

Table 8. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, Equidock, and CNS using predicted inter-chain contacts for CASP-CAPRI dataset. Known tertiary structures are used as monomers.

Methods	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	0.73	11.88	33.21	10.43	26.52
GD	**0.74**	**11.09**	**35.51**	**9.66**	**25.21**
MC	0.72	12.04	32.65	10.47	26.98
CNS	0.62	14.55	28.32	14.37	36.25
Equidock	0.56	18.57	6.51	14.5	35.24

Table 9. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, Equidock, and CNS using predicted inter-chain contacts for CASP-CAPRI dataset. Predicted tertiary structures by AlphaFold2 are used as monomers.

Methods	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	0.64	12.18	27.1	10.73	26.54

GD	**0.69**	**12.15**	**28.05**	**10.69**	**26.50**
MC	0.63	12.78	26.81	11.89	28.92
CNS	0.62	14.55	13.58	12.62	31.69
Equidock	0.50	26.59	22.27	18.39	44.66

Performance evaluation of five methods on 31 heterodimers from Std32

Table 10, 11, and 13 compares the performance of the above-mentioned methods in terms of TM-score, RMSD, f_nat, I_RMSD, and L_RMSD using both true and predicted inter-chain contacts and monomer structures for Std32. Per dimer detailed results are also reported in Tables S10, S11, and S12 in supplementary materials. As shown in Table 10, when native inter-protein contacts are provided as input, DRLComplex has superior performance than all the other methods in terms of all the metrics, except f_nt, for which GD performs better. While DRLComplex achieves a low RMSD of 0.88Å, the models built by GD, MC, and CNS have RMSDs of 2.9Å, 3.1Å, and 10.04Å, respectively. DRLComplex has I_RMSD, and L_RMSD of 0.92Å and 2.15Å, successively, significantly lower than those of the other methods. DRLComplex, GD, and MC perform alike with regard to TM-score.

Using predicted inter-chain contacts, DRLcomplex has higher TM-score and f_nat, and lower I_RMSD, and L_RMSD than GD, MC, CNS, and Equidock. GD achieves a slightly lower RMSD than DRLComplex (RMSD of DRLComplex = 13.93, RMSD of GD = 13.92). GD is able to reconstruct models of higher quality compared to MC, CNS, and Equidock. Equidock has the worst performance in terms of all the metrics.

As can be seen in Table 12, for the realistic scenario, DRLComplex has the best performance regarding TM-score, RMSD, and f_nat, whereas GD achieves the best results for I_RMSD, and L_RMSD. CNS and Equidock have the poorest performance among all the methods. While DRLComplex, GD, and CNS achieved a mean TM-score of about 0.72, TM-scores of the models by CNS, and Equidock are 0.66, and 0.59, respectively. DRLComplex, GD, and MC have an average f_nat of 11.59%, whereas CNS, and Equidock yield an average f_nat of 3.66%.

Similar to the results for homodimers, using predicted inter-chain contacts instead of native inter-chain contacts causes a decrease in the quality of the reconstructed models for heterodimers. Also, as shown in Figures 22, 23, and 24, the quality of the generated models relies on the quality of the predicted inter-chain contacts.

Table 10. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, and CNS using true inter-chain contacts for 31 protein dimers from Std32 dataset. Known tertiary structures are used as monomers.

Methods	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	**0.98**	**0.88**	90.03	**0.92**	**2.15**
GD	0.95	2.9	**92.43**	1.99	7.16
MC	0.94	3.1	92.24	2.2	7.18
CNS	0.82	10.04	69.13	3.71	14.99

Table 11. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, Equidock, and CNS using predicted inter-chain contacts for 31 protein dimers from Std32 dataset. Known tertiary structures are used as monomers.

	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	**0.76**	13.93	**19.64**	**13.68**	**34.16**
GD	0.75	**13.92**	16.68	13.72	34.17
MC	0.73	14.14	13.88	13.82	35.36
CNS	0.68	17.09	17.26	15.81	43.12
Equidock	0.61	18.53	4.95	14.98	36.11

Table 12. Averaged TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of DRLComplex, GD, MC, Equidock, and CNS using predicted inter-chain contacts for 31 protein dimers from Std32 dataset. predicted tertiary structures by AlphaFold2 are used as monomers.

Methods	TM-score	RMSD	F_nat (%)	I_RMSD	L_RMSD
DRLComplex	**0.74**	**14.53**	**11.80**	13.86	36.18
GD	0.73	14.54	11.71	**13.85**	**36.17**
MC	0.7	15.06	11.25	14.60	36.66
CNS	0.66	19.02	3.78	14.74	37.91
Equidock	0.59	18.63	3.55	14.42	35.97

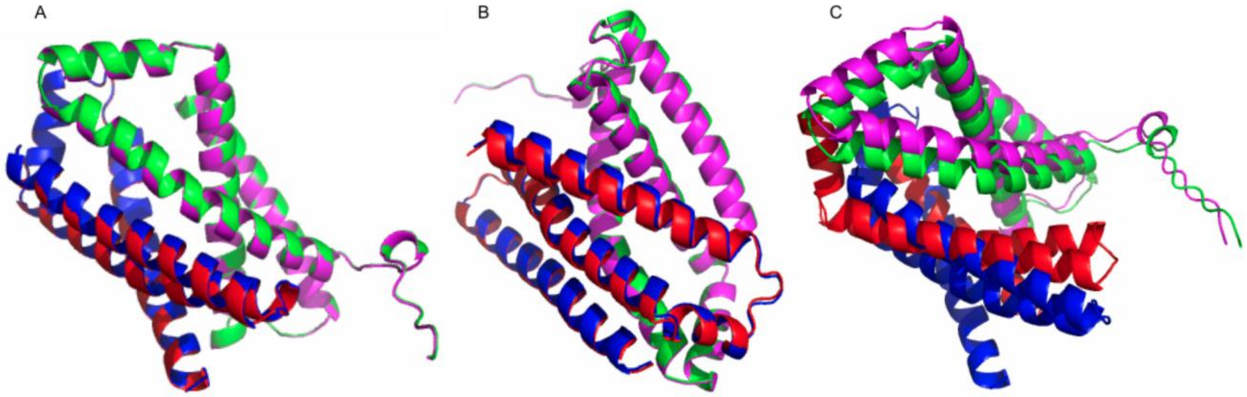

Figure 21. The predicted structures of a heterodimer from Std32 (PDB code: 2WDQ chains C and D) are superimposed on the native structure for optimal, suboptimal, and realistic scenarios. Green and red represent the two chains of the native structure, whereas blue and magenta show the two chains of the predicted structure. (A) The model reconstructed by DRLComplex when native monomer structure and inter-chain contacts are provided as inputs. The detailed results of the generated model are as follow: TM-score = 0.97, RMSD = 0.94Å, f_nat = 97.1%, I_RMSD = 0.95, and L_RMSD = 1.85Å. (B) The model built by DRLComplex using the native monomer structure and predicted inter-chain contacts. TM-score, RMSD, f_nat, I_RMSD, and L_RMSD are 0.99, 0.65Å, 92.1%, 0.58, and 1.12Å, successively. (C) DRLComplex receives predicted monomer structure and inter-protein contacts as inputs and reconstructs a model with TM-score of 0.75, RMSD of 6.03Å, f_nat of 1%, I_RMSD of 6.07, and L_RMSD of 10.6Å.

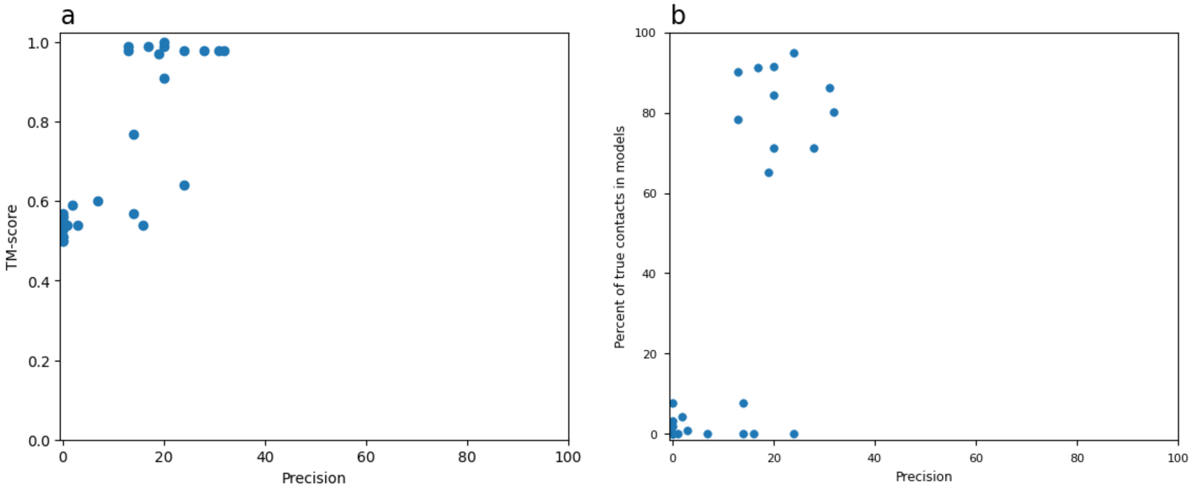

Figure 22. (a) TM-score of the reconstructed models by DRLComplex against the precision of the predicted inter-chain contacts. (b) f_nat values of the generated models versus the precision of the inter-protein contacts. As the precision increases, the quality of the models also increases. The experiment has been performed on the CASP-CAPRI dataset.

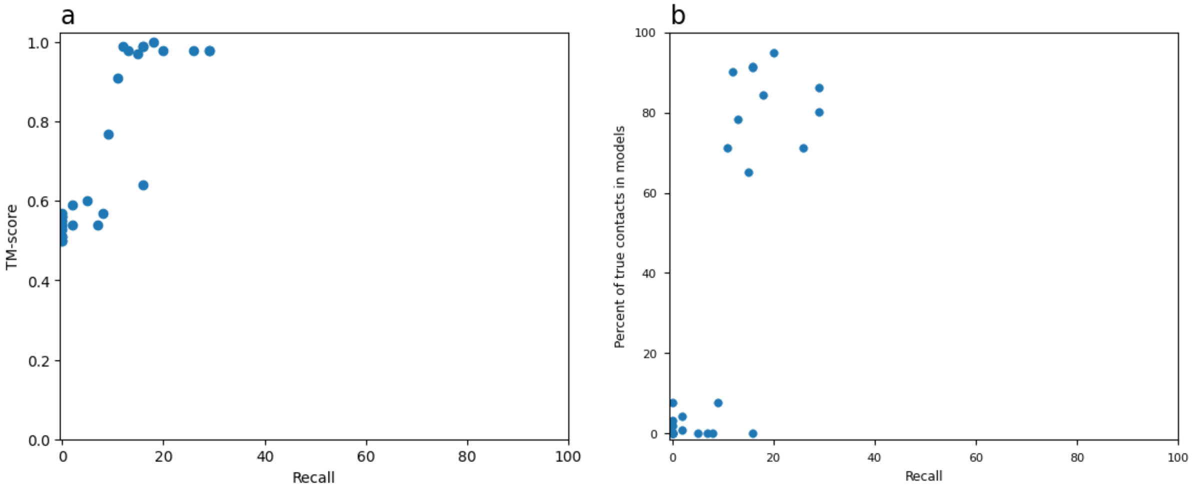

Figure 23. (a) TM-score of the generated models vs the recall on the predicted inter-chain contacts. (b) f_nat of the reconstructed models against the recall of the predicted inter-protein contacts. The quality of the models regarding TM-score and f_nat is dependent on the quality of the predicted contacts. The experiment has been done on the CASP-CAPRI dataset.

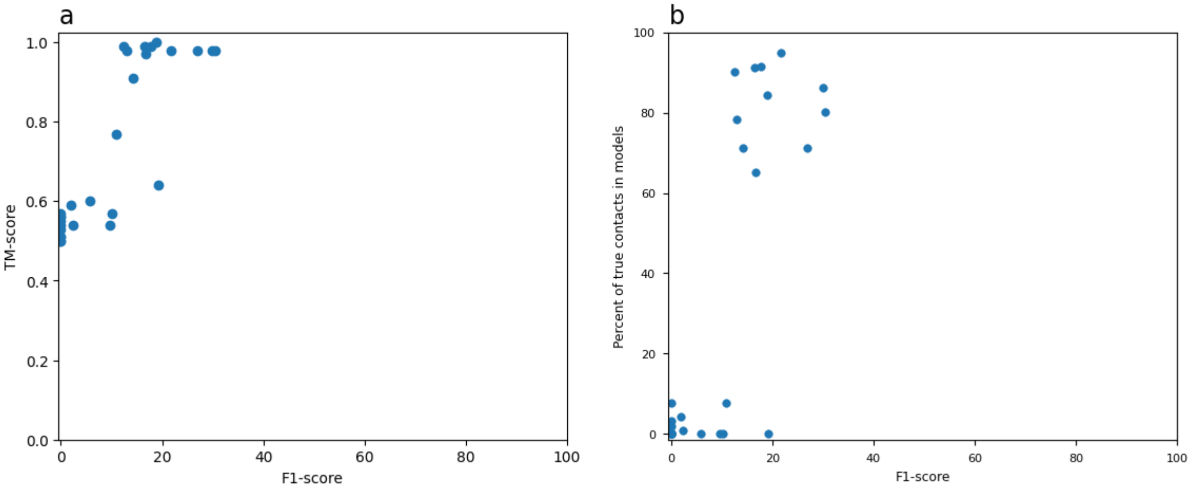

Figure 24. (a) TM-score of the models built by DRLComplex vs the F1-score of the predicted inter-chain contacts. (b) f_nat values of the models reconstructed by DRLComplex vs the F1-score of the predicted inter-protein contacts. The experiment has been done on the CASP-CAPRI dataset.

We also use GD to initialize the starting conformation for DRLComplex. The results for Std32 using predicted inter-chain contacts and true monomers are shown in Table 13. Also, Figure 25 shows the TM-score of the models generated by GD and those of DRLComplex initialized by GD. The average TM-score of the best models reconstructed by DRLComplex is 0.76, 0.1 higher than that of GD. It is worth noting that compared to GD, DRLComplex always satisfies a higher portion of restraints. However, since restraints are noisy and conflicting, a higher portion of satisfied constraints does not always lead to a higher quality model (see Figure 25).

Table 13. Detailed results (TM-score, RMSD, F_nat, I_RMSD and L_RMSD) of DRLComplex for Std32 using predicted inter-chain contacts and true monomers as inputs. The model reconstructed by GD is used as initial conformation for DRLComplex.

Target	**TM-score**	**RMSD**	**F_nat**	**IRMSD**	**LRMSD**
1EFP,AB	0.85	3.02	25	3.16	7.69
1EP3,AB	0.8	7.96	17	6.41	17.8
1I1Q,AB	0.71	19.38	8	21.3	55.57
1QOP,AB	0.63	20.43	4	23.73	51.2

1W85,AB	0.94	0.69	80	0.54	1.99
1ZUN,AB	0.77	15.37	2	16.47	30.14
2D1P,BC	0.56	14.72	0	14.07	34.28
2NU9,AB	0.92	2.53	54	2.15	5.02
2ONK,AC	0.5	27.56	2	25.44	65.55
2VPZ,AB	0.84	34.43	6	32.18	92.48
2WDQ,CD	1	0.66	92	0.64	1.16
2Y69,AB	0.72	26	7	27.28	65.07
2Y69,AC	0.72	25.52	6	25.12	59.67
2Y69,BC	0.53	21.02	1	16.55	65.58
3A0R,AB	0.67	17.59	2	20.53	43.79
3G5O,AB	0.58	8.94	9	9.46	13.95
3IP4,AB	0.52	25.54	6	20.57	57.88
3IP4,AC	0.81	12.26	3	14.94	31.51
3IP4,BC	1	0.76	74	0.56	2.23
3MML,AB	0.88	2.78	23	2.47	5.71
3OAA,HG	0.87	3.87	29	4.38	8
3PNL,AB	0.78	6.85	16	6.2	17.01
3RPF,AC	0.62	17.53	6	9.12	53.98

3RRL,AB	0.69	12.96	3	12.69	25.21
4HR7,AB	0.96	8.33	15	9.68	23.35
1B70,AB	0.74	16.81	7	20.74	34.83
1BXR,AB	0.75	20.76	15	20.85	49.95
1RM6,AB	0.68	26.2	9	26	59.24

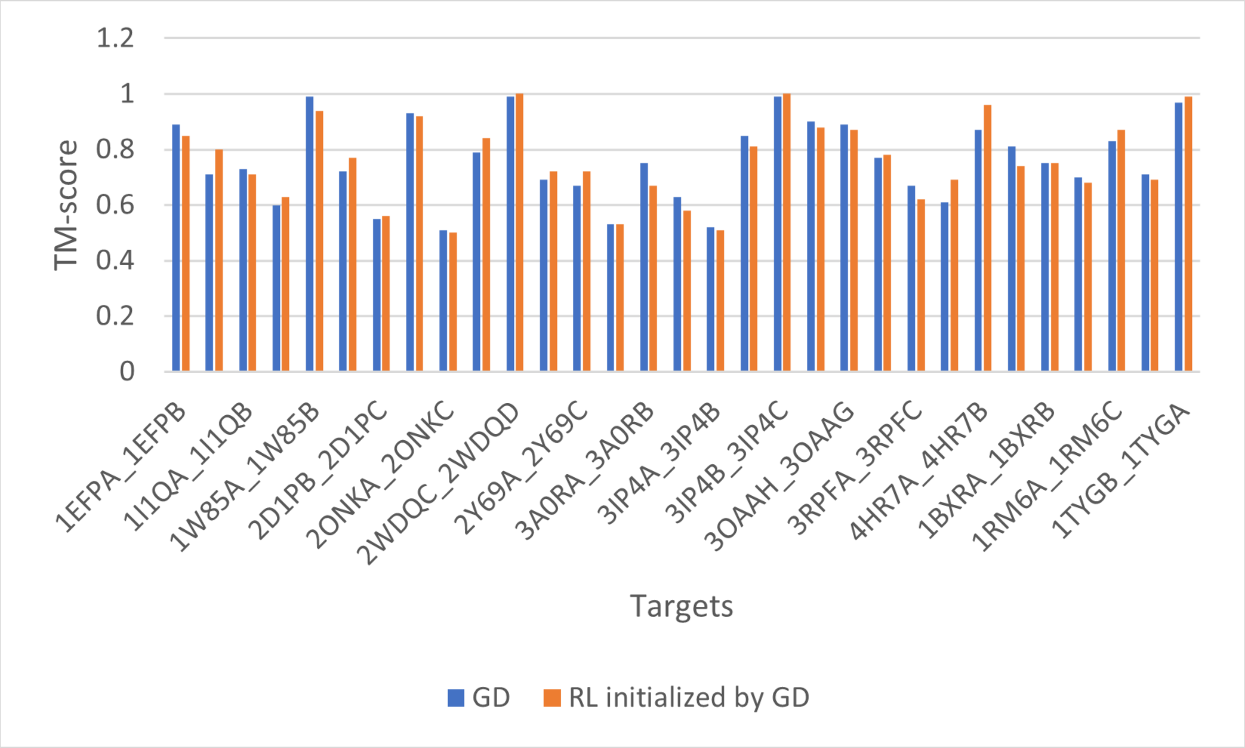

Figure 25. GD vs RL initialized by GD. For 15 out of 31, RL improved the TM-score of the reconstructed models by GD.

The running time of the algorithms is shown in Table 14. As can be seen in Table 14, starting from the model reconstructed by GD reduced the running time of DRLComplex by 22 %. Based on the table, CNS is the slowest method to reconstruct quaternary structures. Although the least accurate method, Equidock is the fastest algorithm.

Table 14. Running time of the algorithms. GD, MC, and CNS generate 100 different models.

Method	*Time/Sequence Length*
GD	30 mins / 1045
DRLComplex	4.5 hours / 1045
DRLComplex initialized with GD	2 hours / 1045
MC	6 hours / 1045
CNS	22 hours / 1045
Equidock	2 mins / 1045

Conclusion

We propose an optimization method based on gradient descent, called GD, to build protein complexes using inter-chain contacts as restraints. We compare the performance of GD with a Markov Chain Monte Carlo method (MC), and a method based on simulated annealing (CNS). GD has the best performance among the three methods for building protein dimers using true and predicted inter-protein contacts. Not only is GD able to build high quality models for nearly all homodimers and heterodimers when native inter-chain contacts are provided, but it also reconstructs good quality models for many protein complexes using only predicted contacts.

We also introduce an agent-based reinforcement learning system (DRLComplex), which learns to reconstruct protein dimers from true/predicted contacts through a self-learning process. DRLComplex rotates and translates the ligand with respect to the receptor in the translation/rotation space using not only immediate rewards but also long-term rewards, with the main aim of detecting a conformation as close to the native structure as possible. Using native inter-protein contacts as restraints, DRLComplex generates high quality models for all the protein dimers. In addition, if predicted contacts are provided as inputs, DRLComplex builds models with mean TM-scores of 0.73, and 0.76 for CASP-CAPRI dataset (28 homodimers) and Std32 (32 heterodimers), respectively, indicating that the method can generates models with reasonable quality utilizing only predicted contacts. Furthermore, DRLComplex's performance is comparable to GD for homodimers; however, this self-learning method outperforms all the other methods for heterodimers.

Supplementary Data

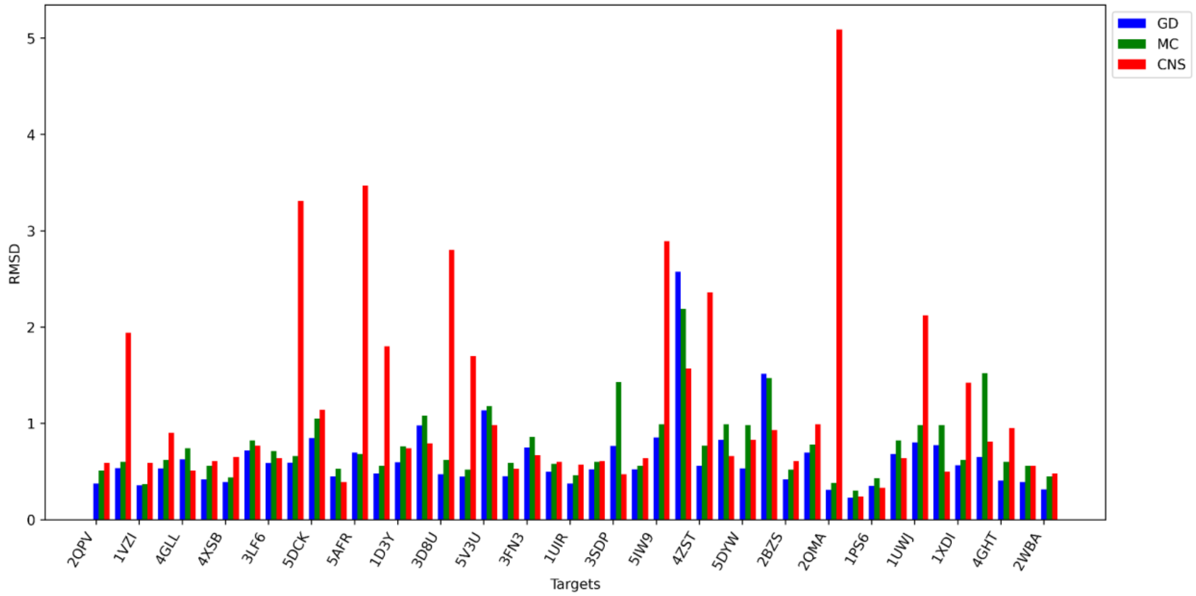

Figure S1. RMSD of GD, MC, and CNS on a dataset of 44 homodimers with known inter-protein contacts. The average RMSD of GD, MC, and CNS is 0.63, 0.76, and 1.16.

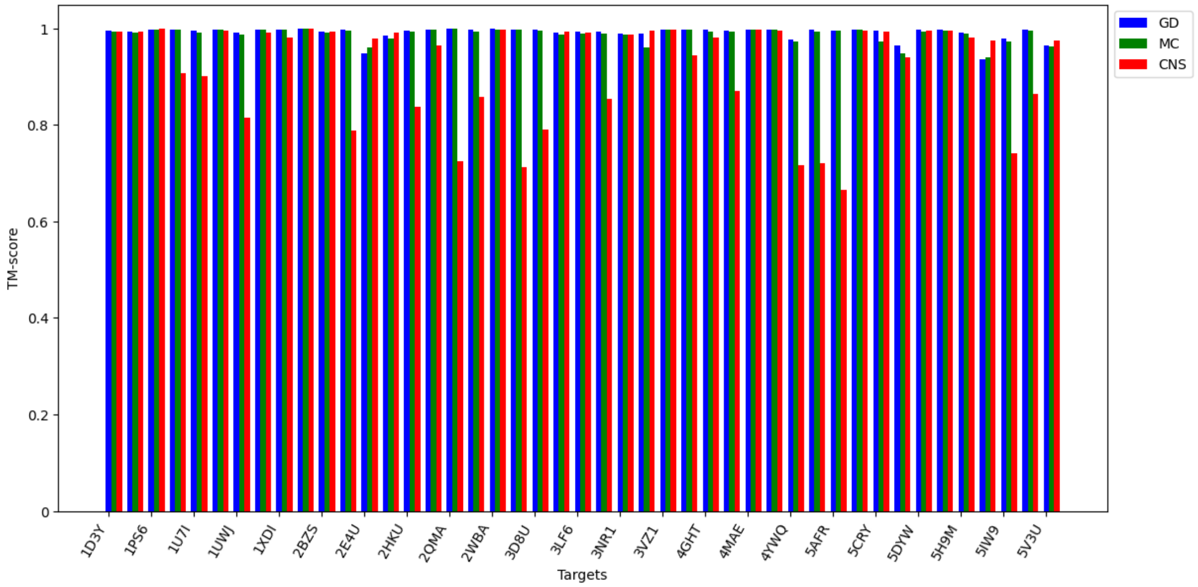

Figure S2. TM-score of GD, MC, and CNS on a dataset of 44 protein complexes with known inter-protein contacts. The average TM-score of GD, MC, and CNS is 0.99, 0.98, and 0.91.

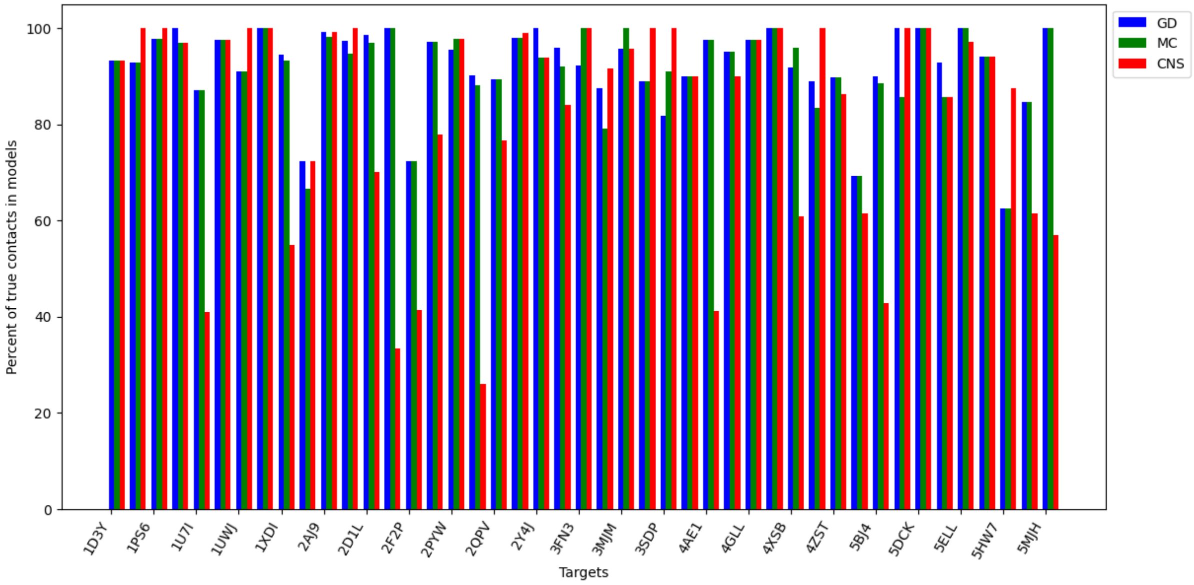

Figure S3. The f_nat of GD, MC, and CNS on a dataset of 44 homodimers with known inter-protein contacts. The average f_nat of GD, MC, and CNS is 92.19, 91.39, and 82.49.

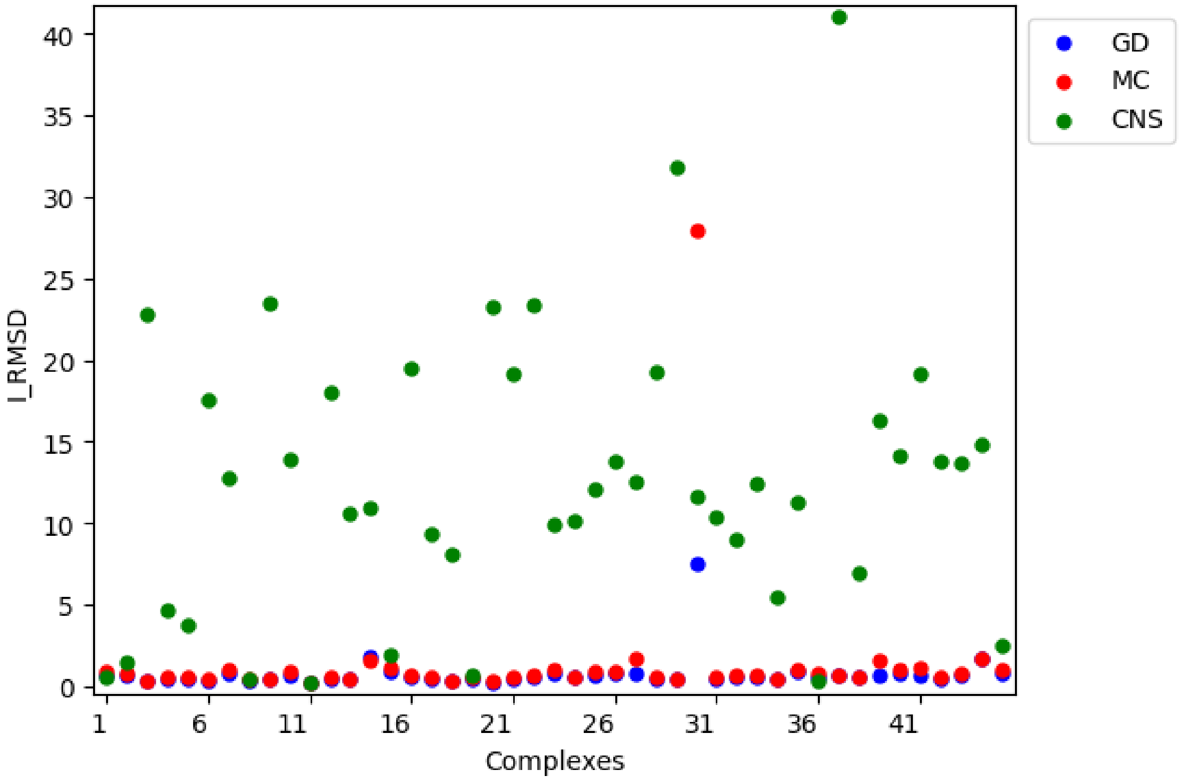

Figure S4. I_RMSD of GD, MC, and CNS on a dataset of 44 homodimers with known inter-protein contacts. The average I_RMSD of GD, MC, and CNS is 0.77, 1.35, and 12.46.

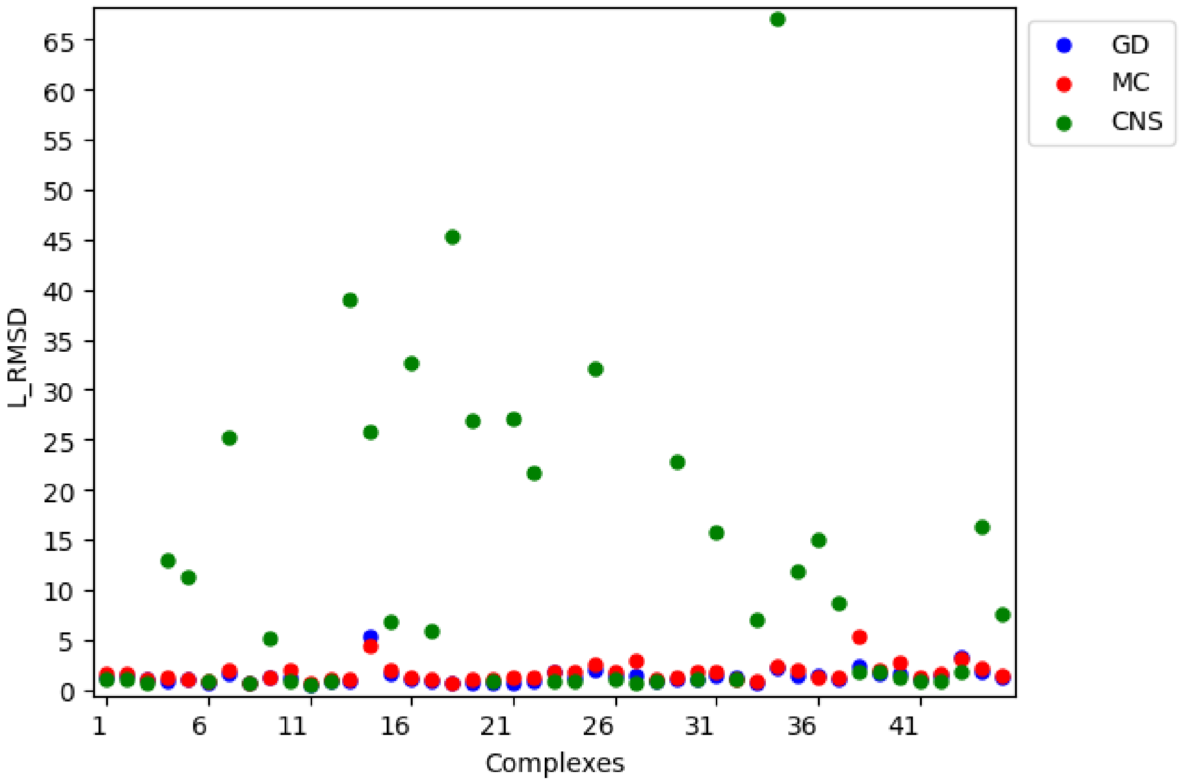

Figure S5. L_RMSD of GD, MC, and CNS on a dataset of 44 homodimers with known inter-protein contacts. The average L_RMSD of GD, MC, and CNS is 1.38, 1.7, and 11.18.

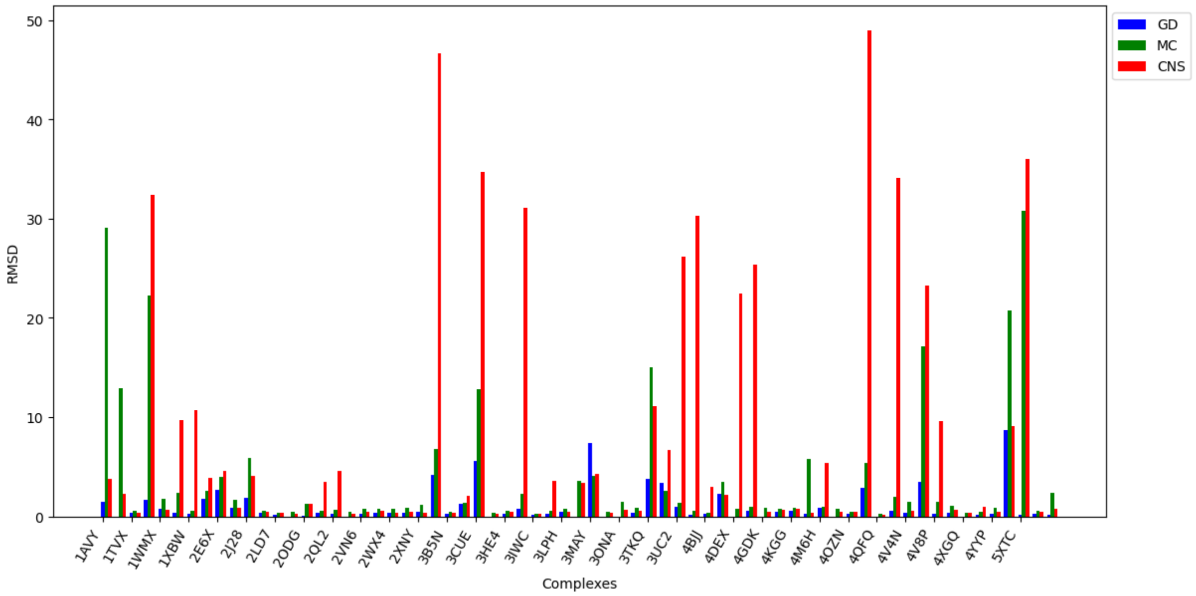

Figure S6. RMSD of GD, MC, and CNS on 73 heterodimers with known inter-protein contacts. The average RMSD of GD, MC, and CNS is 1.23, 4.76, and 7.7.

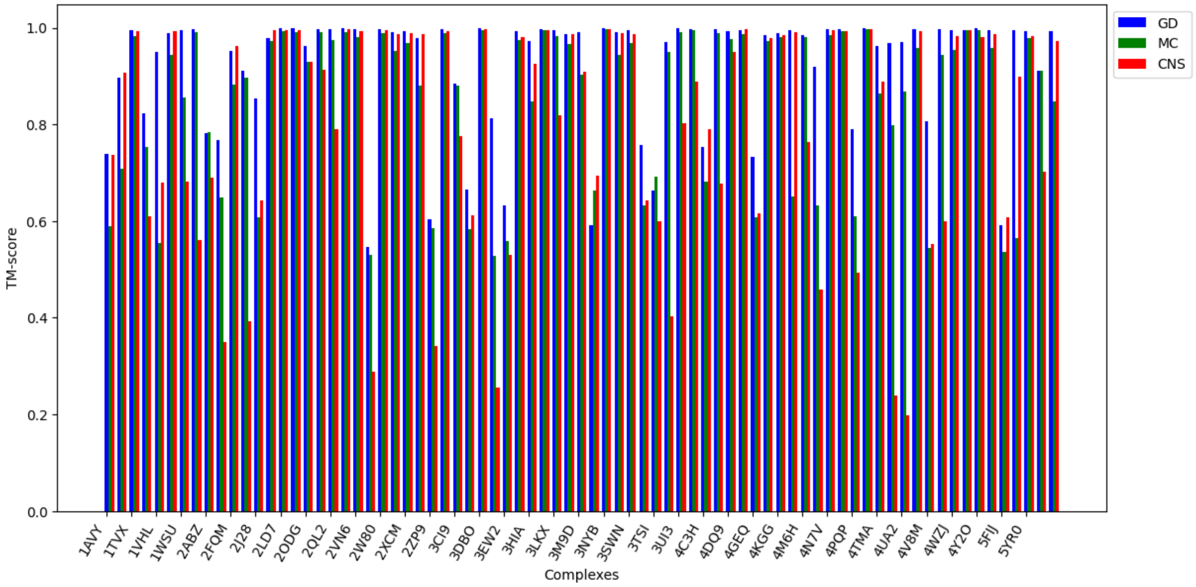

Figure S7. TM-score of GD, MC, and CNS on 73 heterodimers with known inter-protein contacts. The average TM-score of GD, MC, and CNS is 0.92, 0.85, and 0.79.

Table S1. TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of the models that GD reconstructed for each of 44 homodimers from true inter-chain contacts.

Target	Number of true contacts	Length of chain A	Length of chain B	TM-score	RMSD	f_nat (%)	I_RMSD	L_RMSD
2QMA	621	444	440	0.999	0.31	90.2	0.325	0.619
2E4U	52	512	514	0.948	2.576	100	1.85	5.314
5DCK	53	71	72	0.965	0.846	100	0.754	1.674
5CRY	59	348	348	0.995	0.652	100	0.682	2.315
5IW9	67	123	122	0.979	0.853	84.6	0.842	1.727
4GHT	77	181	181	0.997	0.407	95	0.41	1.128
3SDP	78	186	186	0.988	0.764	81.8	0.813	1.438
5V3U	83	131	123	0.965	1.135	90.9	1.12	2.64
5DYW	84	527	525	0.998	0.532	92.9	0.62	1.682
5AFR	88	327	325	0.994	0.696	69.19	0.681	1.349
4YWQ	89	147	146	0.977	0.979	88.9	0.847	2.19
3FN3	94	215	211	0.99	0.748	92.3	0.84	1.737
2Y4J	99	377	377	0.998	0.391	100	0.398	0.788
5H9M	103	190	189	0.99	0.696	94.1	0.642	1.312
3D8U	107	260	266	0.997	0.471	96	0.719	1.39

3LF6	110	154	157	0.992	0.588	87.5	0.526	0.949
1UWJ	111	264	263	0.99	0.802	90.9	0.564	1.247
1JCZ	116	260	260	0.993	0.683	92.9	0.833	1.603
3NR1	130	178	178	0.989	0.717	88.9	0.674	1.43
3MJM	132	343	342	0.993	0.773	95.7	0.748	1.512
4GLL	139	307	306	0.995	0.625	97.5	0.657	2.08
5ELL	142	231	235	0.996	0.499	100	0.558	1.423
1PS6	144	328	328	0.998	0.348	97.7	0.438	1.108
4ZST	146	328	328	0.996	0.56	89.7	0.281	0.999
2HKU	152	188	182	0.994	0.534	97.2	0.569	1.358
2F2P	155	169	169	0.985	0.827	72.39	0.603	1.047
4MAE	161	577	577	0.998	0.522	100	0.924	1.598
1U7I	164	130	129	0.994	0.48	87.2	0.601	1.219
1SOX	167	463	458	0.998	0.452	100	0.492	0.975
2QPV	169	128	128	0.996	0.375	89.4	0.46	0.964
1VZI	174	125	125	0.996	0.355	100	0.417	0.729
1UIR	179	309	313	0.998	0.376	97.6	0.373	0.717
2PYW	197	417	411	0.998	0.448	95.6	0.377	0.763

2BZS	207	230	228	0.993	0.419	97.3	0.45	0.949
2D1L	236	240	222	0.997	0.447	98.5	0.441	0.859
1D3Y	250	289	290	0.995	0.597	93.3	0.423	0.914
3VZ1	273	452	452	0.998	0.417	90.0	0.671	1.341
5MJH	320	368	368	0.996	0.593	100	0.418	0.844
4XSB	323	340	343	0.998	0.39	91.9	0.65	1.261
5BJ4	340	366	366	0.997	0.524	90	0.403	0.784
2WBA	347	489	489	0.999	0.313	98	0.535	1.049
1XDI	352	459	459	0.997	0.562	94.5	0.249	0.747
2AJ9	364	334	334	0.999	0.228	99.1	0.48	1.225
4AE1	398	501	501	0.997	0.531	97.5	0.204	0.446
5HW7	39	122	119	0.936	1.515	62.5	0.419	1.061

Table S2. TM-score, RMSD, f_nat, I_RMSD, and L_RMSD of structural models reconstructed by GD for 73 heterodimers in the Hetero73 dataset using true/native inter-chain contacts as input.

Target	Chains	Length of chain 1	Length of chain 2	Number of true contacts	TM-score	RMSD	f_nat	I_RMSD	L_RMSD
1AVY	A, B	68	54	19	0.74	1.47	92.9	0	6.91
1IS7	A, K	194	84	10	0.9	0	85.7	1.61	4.97
1TVX	A, B	64	71	86	0.99	0.33	100	0.35	0.66
1VCH	B, E	170	152	19	0.82	3.33	42.9	1.76	9.22
1VHL	B, C	208	183	22	0.95	1.65	72.7	1.53	4.04
1WMX	A, B	173	195	61	0.99	0.74	96.89	0.49	1.75
1WSU	C, D	102	121	49	0.99	0.39	100	0.38	1.21
1XBW	A, B	99	96	149	1	0.26	97.7	0.25	0.63
2ABZ	C, D	62	46	13	0.78	1.74	54.5	2	5.06
2E6X	C, D	56	66	23	0.77	2.64	85.7	0	5.38
2FQM	A, D	65	72	21	0.95	0.84	100	0.76	3.57
2IS5	A, D	156	143	43	0.91	2.01	61.9	1.88	4.04
2J28	1, 3	54	64	11	0.85	1.82	83.3	0.93	6.05
2JG8	A, B	132	129	119	0.98	0.32	98.6	0.32	0.67
2LD7	A, B	94	75	153	1	0.18	95.7	0.18	0.34

2MJF	A, B	40	95	132	1	0	97.8	0.16	0.3
2ODG	A, C	89	47	30	0.96	0.07	78.6	0	1.65
2P7M	A, B	127	122	199	1	0.33	97	0.33	0.65
2QL2	A, B	56	59	79	1	0.25	94.69	0.23	0.58
2ROZ	A, B	32	136	101	1	0	100	0.21	0.46
2VN6	A, B	151	64	80	1	0.27	100	0.3	0.61
2W80	A, D	123	244	145	1	0.39	100	0.42	0.87
2WX4	B, C	41	43	61	0.99	0.3	100	0.3	0.6
2XCM	C, E	92	74	57	0.99	0.4	100	0.37	0.81
2XNY	M, N	37	36	77	0.98	0.44	83.6	0.43	0.87
2ZP9	E, I	49	39	4	0.6	4.17	100	3.06	16.47
3B5N	B, C	69	70	149	1	0.23	100	0.23	0.45
3CI9	A, B	44	45	20	0.88	1.28	100	0.54	4.62
3CUE	B, Q	167	188	5	0.67	5.6	50	0.98	23.58
3DBO	A, B	34	126	171	1	0	98.1	0.1	0.44
3ERM	D, E	63	56	16	0.81	2.14	81.8	0.43	6.61
3EW2	C, F	124	119	10	0.63	5.36	85.7	2.06	20.79
3HE4	A, B	45	44	90	0.99	0.28	100	0.25	0.53

3HIA	B, C	83	74	38	0.97	0.8	88.5	0.69	1.72
3IWC	A, B	58	61	255	1	0.18	96.89	0.18	0.36
3LKX	A, B	65	53	123	1	0.28	100	0.28	0.54
3LPH	A, B	62	55	65	0.99	0.44	100	0.44	0.97
3M9D	A, G	186	31	29	0.99	0	90.9	0.71	1.55
3MAY	A, C	86	97	6	0.59	7.39	83.3	2.78	15.12
3NYB	A, B	323	64	173	1	0	99.2	0.27	0.57
3ONA	A, B	158	66	61	0.99	0	97.1	0.46	1.22
3SWN	R, S	76	72	98	1	0.32	100	0.32	0.66
3TKQ	A, E	191	166	5	0.76	3.8	50	4.74	10.04
3UC2	B, D	125	109	43	0.97	0.98	88.5	0.76	2.63
3UI3	A, B	142	102	177	1	0.17	99.1	0.17	0.39
4BJJ	A, B	106	85	192	1	0.28	96.5	0.29	0.56
4C3H	J, L	69	45	18	0.75	2.26	90	0	7.34
4DEX	A, B	289	45	93	1	0	96.2	0.38	1.03
4DQ9	A, B	149	141	66	0.99	0.51	98	0.49	1.02
4GDK	A, B	88	267	71	0.99	0	95.8	0.37	1.96
4GEQ	B, C	58	90	6	0.73	2.61	50	1.57	15.37

4K12	A, B	64	82	57	0.98	0.5	100	0.45	1.71
4KGG	D, A	163	141	66	0.99	0.57	96.2	0.56	1.16
4M3L	A, D	60	53	59	0.99	0.28	100	0.3	0.67
4M6H	A, B	190	162	66	0.99	0.84	97.1	0.59	2.14
4M77	H, J	85	72	23	0.92	1.46	100	0.56	3.81
4N7V	A, C	222	33	98	1	0	100	0.2	0.76
4OZN	A, B	116	104	124	1	0.28	98.8	0.29	0.68
4PQP	A, D	102	97	17	0.79	2.89	64.3	1.24	8.39
4QFQ	A, B	101	35	250	1	0	98.4	0	0.29
4TMA	I, J	47	57	42	0.96	0.6	93.5	0.57	1.83
4U3Q	A, B	93	99	17	0.97	0.94	81.8	0.94	2.62
4UA2	A, H	115	103	27	0.97	0.95	94.1	0.59	2.08
4V4N	T, W	215	135	107	1	0.31	100	0.35	0.85
4V8P	K, M	108	143	65	1	0.28	100	0.28	0.95
4WZJ	L, M	79	79	98	0.99	0.34	98.1	0.32	0.64
4XGQ	A, B	132	30	90	1	0	98.5	0.36	0.77
4Y2O	A, B	211	142	170	1	0.19	98	0.2	0.44
4YYP	A, B	87	32	86	0.99	0.3	100	0.33	0.71

5FIJ	S, T	167	174	7	0.59	8.68	11.4	0	36.13
5XTC	B, V	124	111	64	0.99	0.17	97.1	0.44	0.61
5YR0	A, B	48	44	75	0.99	0.3	100	0.29	0.58
6UMM	D, I	81	61	23	0.99	0.11	100	0.48	0.68

Table S3. Average TM-score, RMSD, f_nat, I_RMSD, and L_RMSD) of GD on 32 heterodimers in the Std32 dataset using true contacts as input.

Target	**Length of chain 1**	**Length of chain 2**	**Chains**	**Number of true contacts**	**TM-score**	**RMSD**	**f_nat**	**I_RMSD**	**L_RMSD**
1W85	358	324	A, B	185	1	0.11	100	0.1	0.22
1EFP	307	246	A, B	317	1	0.07	97.8	0.08	0.18
1I1Q	512	186	A, B	153	1	0.08	100	0.08	0.18
2Y69	227	259	B, C	2	0.54	13.88	50	10	11.01
3MML	285	207	A, B	146	1	0.12	96.3	0.12	0.38
2VPZ	734	193	A, B	188	1	0.19	94.1	0.24	0.58
1TYG	65	242	B, A	114	1	0.05	98.7	0.08	0.26
3RPF	143	72	A, C	5	0.82	3.49	100	1.15	7.44
1EP3	311	261	A, B	133	1	0.05	100	0.04	0.12

2NU9	285	385	A, B	190	1	0.11	98.6	0.11	0.21
3RRL	227	197	A, B	194	1	0.14	99.3	0.14	0.45
3IP4	485	482	A, B	183	1	0.12	98.3	0.02	0.24
1RM6	761	323	A, B	129	1	0.18	100	0.1	0.28
2D1P	119	95	B, C	60	1	0.12	97.3	0.1	0.33
4HR7	443	80	A, B	36	0.99	0.78	100	0.35	2.2
2ONK	240	252	A, C	92	1	0.32	92.6	0	0.9
3A0R	334	113	A, B	60	1	0.28	100	0.06	0.71
1B70	265	775	A, B	414	1	0.07	95.7	0.08	0.23
1QOP	265	390	A, B	157	1	0.15	99	0.15	0.32
2WDQ	121	105	C, D	77	1	0.12	100	0.12	0.27
1BXR	1073	379	A, B	258	1	0.07	98.7	0.07	0.17
3G5O	92	81	A, B	143	1	0.05	97.8	0.05	0.08
3OAA	138	284	H, G	240	1	0.11	98.3	0.11	0.33
3PNL	356	211	A, B	148	1	0.2	100	0.19	0.55
1ZUN	196	382	A, B	251	1	0.14	100	0.15	0.45
1IXR	135	308	A, C	0	0.69	29.9	0	29	92.99
1W85	358	324	A, B	185	1	0.11	100	0.1	0.22

1EFP	307	246	A, B	317	1	0.07	97.8	0.08	0.18
1I1Q	512	186	A, B	153	1	0.08	100	0.08	0.18
2Y69	227	259	B, C	2	0.54	13.88	50	10	11.01
3MML	285	207	A, B	146	1	0.12	96.3	0.12	0.38
2VPZ	734	193	A, B	188	1	0.19	94.1	0.24	0.58

Table S4. Detailed results of GD on Set A with predicted inter-chain contacts as input.

Target name	**Length of chain A**	**Length of chain B**	**Number of predicted interchain contacts**	**Precision of predicted interchain contacts (%)**	**Recall of predicted inter-chain contacts (%)**	**TM-score**	**RMSD**	**f_nat**	**I_RMSD**	**L_RMSD**
2XBQ	105	105	26	0.0	0.0	0.5	14.74	0.0	14.544	43.547
1Z9Z	60	60	32	0.0	0.0	0.55	11.27	0.0	13.191	20.364
1A19	89	89	26	0.0	0.0	0.5	14.96	0.0	14.706	39.503
1YH8	266	266	54	0.0	0.0	0.5	18.47	0.0	17.312	58.735
3N8E	159	159	53	0.0	0.0	0.51	21.39	0.0	22.79	44.298

5LLJ	57	57	41	3.12	8.0	0.53	13.20	0.0	9.685	22.742
2FU4	81	81	29	0.0	0.0	0.5	17.39	0.0	11.888	38.914
2PL7	67	67	25	1.85	3.33	0.51	10.19	0.0	9.634	28.831
4E83	31	31	41	5.08	9.68	0.55	9.36	12.5	6.373	13.754
5UZX	231	231	48	0.0	0.0	0.5	22.67	0.0	24.152	66.521
1A2D	130	130	53	2.35	5.88	0.67	6.44	10	6.57	11.312
1RRG	177	177	62	0.0	0.0	0.64	8.71	0.0	7.635	16.947
3JSL	308	308	29	0.0	0.0	0.5	23.37	0.0	24.824	55.067
3LO2	30	30	43	37.5	50	0.86	1.1	75	1.048	2.201
4Q1R	130	130	24	46.34	52.78	0.98	0.62	84.6	0.614	1.78
1VH9	138	138	82	3.48	10.81	0.5	13.97	16.7	13.865	35.895
1D8U	165	165	48	8.86	18.42	0.6	7.48	0.0	7.45	17.776
4GA9	134	134	58	54.69	85.37	0.9	2.06	57.1	2.083	3.787
1IU8	206	206	38	8.97	14.89	0.61	18.85	0.0	16.81	26.005
2R74	142	142	61	0.0	0.0	0.5	15.99	0.0	15.796	45.492
5C39	51	51	90	34.67	52	0.99	0.26	92.3	0.196	0.518
1F86	115	115	82	32.67	63.46	0.99	0.36	100	0.397	0.724
2ZWM	120	120	21	1.39	1.92	0.62	11.58	0.0	11.444	18.365

1M0U	203	203	94	46.08	85.45	0.99	0.7	100	0.766	1.427
1PD3	54	54	28	1.22	1.82	0.94	0.94	20.8	1.2	1.98
2D4G	165	165	98	2.68	7.27	0.6	19.53	0.0	12.094	28.873
3F08	135	135	113	0.58	1.67	0.53	16.71	0.0	17.982	32.181
2CC3	144	144	28	2.3	3.28	0.62	9.01	0.0	7.978	15.46
1GNW	210	210	34	28	33.87	0.99	0.54	60.9	0.562	1.046
2CCY	127	127	53	11.65	19.35	0.64	17.74	0.0	19.329	25.58
5F5X	333	333	30	13.41	17.46	0.7	5.38	0.0	5.183	11.139
5JYB	344	344	37	14.77	20.31	0.97	1.61	11.8	1.668	4.962
1EOG	208	208	99	51.85	86.15	0.99	0.29	96.2	0.311	0.629
1MK4	157	157	24	0.0	0.0	0.6	12.29	0.0	12.115	20.927
1HNB	217	217	58	40.45	53.73	0.99	0.61	81	0.526	1.348
2CVI	83	83	40	5.94	8.96	0.62	6.19	0.0	6.193	11.909
1V8F	276	276	35	17.05	22.06	0.5	26.92	0.0	21.932	56.044
1YQ1	198	198	51	36.78	47.06	1	0.11	95.7	0.118	0.214
3BBH	204	204	22	8.43	10.29	0.92	2.11	29.4	2.423	4.107
3RHU	141	141	49	0.0	0.0	0.5	21.20	0.0	22.02	57.47

Table S5. Detailed results of GD on Set B with predicted inter-chain contacts as input.

Target name	Length of chain A	Length of chain B	Number of predicted interchain contacts	Precision of predicted interchain contacts (%)	Recall of predicted inter-chain contacts (%)	TM-score	RMSD	f_nat	I_RMSD	L_RMSD
1LBK	208	208	94	52.34	81.16	0.99	0.26	96.6	0.27	0.56
2YYB	242	242	31	3.06	4.29	0.5	25.28	0.0	23.48	53.45
1T92	108	108	29	3.06	4.17	0.64	5.97	0.0	5.58	14.56
2QY6	244	244	28	0.0	0.0	0.5	25.84	0.0	21.64	72.11
1ML6	219	219	72	51.04	67.12	1	0.1	96.6	0.12	0.22
5AIF	124	124	22	6.52	7.89	0.94	1.36	24	1.38	2.80
2YR1	257	257	26	8.42	10.39	0.96	1.51	22.2	1.59	4.12
1B48	221	221	68	55.32	66.67	0.99	0.24	100	0.26	0.5
3F1V	366	366	39	27.17	32.05	0.99	0.77	56	0.76	2.47
3KXO	198	198	86	47.75	67.95	0.99	0.28	100	0.29	0.53
1ECS	120	120	94	25.38	44.3	0.97	0.91	71.4	0.80	1.90
3EE2	198	198	90	46.69	68.35	0.99	0.33	92	0.37	0.74
3WVA	163	163	24	1.92	2.44	0.92	1.98	13.6	1.99	4.58
4Q97	108	108	39	3.42	4.88	0.64	19.77	0.0	6.71	25.11

4DBH	269	269	26	7.92	9.64	0.75	5.51	0.0	6.13	12.03
2FHE	216	216	84	47.46	62.22	0.99	0.51	93.8	0.58	1.1
1DUG	234	234	35	35.11	35.87	0.98	0.83	61.3	0.83	1.93
3SW1	134	134	57	0.0	0.0	0.57	12.93	0.0	13.37	21.55
3MMH	167	166	26	2.5	3.09	0.57	13.97	0.0	13.47	22.6
4RAZ	134	134	52	26.27	31.96	0.59	9.55	5	7.02	21.66
3GW7	215	215	23	0.0	0.0	0.5	24.05	0.0	24.7	55.64
1VRW	289	289	25	2.4	2.91	0.58	12.4	0.0	12.98	31.98
4EC7	108	108	22	0.81	0.97	0.54	12.89	2.9	12.4	23.7
2C2X	280	280	40	6.62	8.57	0.97	1.22	22.2	1.34	2.41
1VJ2	114	114	27	3.85	4.63	0.98	0.65	15.2	0.67	1.32
4EP4	166	166	27	5.47	6.48	0.56	15.9	0.0	14.81	29.29
1Z3A	156	156	23	7.32	8.26	0.92	1.84	11.4	1.91	3.7
4ZBD	219	219	84	20.61	24.55	0.98	1.06	42.6	0.76	2.2
1PM7	199	199	105	30.95	45.22	0.98	0.91	55.9	0.78	1.87
2JL4	212	212	71	30.14	36.97	0.97	1.1	65.7	1.17	2.44
3NYG	93	93	99	40.4	51.26	0.99	0.51	69.8	0.53	1.03
2HIQ	96	96	29	2	2.42	0.58	16.31	0	11.67	22.65

1Q7G	358	358	43	7.64	9.52	0.97	1.51	12.5	1.61	2.97
1ITU	369	369	27	5.48	6.3	0.74	6.33	1.9	6.15	14.41
3ZJL	191	191	28	6.16	7.09	0.84	3.28	2	3.05	7.66
1DC4	323	323	38	5.7	6.98	0.51	22.53	0.0	20.69	56.98
1SW7	245	245	154	42.21	65.12	0.99	0.31	88.4	0.31	0.68

Table S6. Detailed results of GD on Set C with predicted inter-chain contacts as input.

Target name	**Length of chain A**	**Length of chain B**	**Number of predicted interchain contacts**	**Precision of predicted interchain contacts (%)**	**Recall of predicted interchain contacts (%)**	**TM-score**	**RMSD**	**f_nat**	**I_RMSD**	**L_RMSD**
3KRS	249	249	88	37.74	45.8	0.975	1.287	56	1.325	2.726
1WYI	248	248	141	46.28	64.93	1	0.159	88.9	0.166	0.375
3OGQ	112	112	21	5.37	5.88	0.525	14.553	0	12.173	31.829
1C6X	99	99	151	40.24	49.28	0.994	0.403	84.6	0.4	0.915
2BTM	250	250	121	38.5	52.17	0.998	0.392	67.3	0.433	0.825
4LUL	189	189	60	22.84	26.62	0.981	0.975	37.8	1.026	2.47
2YPI	247	247	107	40.34	50.71	0.995	0.569	65.9	0.595	1.174

3MWS	99	99	140	38.95	47.86	0.994	0.392	76.8	0.375	1.021
2FDE	99	99	157	38.73	47.52	0.993	0.435	77.8	0.418	0.896
3EM6	99	99	149	39.31	47.89	0.995	0.39	80.4	0.376	0.913
3LZU	99	99	157	40.46	48.61	0.991	0.492	84.5	0.466	0.94
3S45	99	99	147	45.51	52.78	0.991	0.501	84	0.439	1.014
4M8Y	100	100	142	39.43	47.92	0.991	0.492	87.5	0.439	1.056
2AOG	99	99	150	41.04	48.97	0.993	0.44	77.4	0.392	0.956
4M8X	99	99	161	40.8	48.63	0.993	0.446	86.2	0.391	0.974
3U7S	99	99	166	37.43	45.58	0.991	0.504	82.7	0.426	1.112
4YMZ	250	250	104	40	48.65	0.988	0.876	78.9	0.881	2.044
2FDD	99	99	160	40.11	47.65	0.99	0.538	81.8	0.492	1.27
4COB	206	206	38	5.98	7.01	0.602	16.347	0	11.312	24.4
3SK2	132	132	44	16.2	17.68	0.587	18.453	0	11.871	27.072
3DSB	146	146	34	3.11	3.64	0.557	11.259	1.9	11.62	25.167
1KPB	113	113	48	16.22	17.96	0.728	4.262	1.6	4.578	7.736
5CPG	155	155	97	2.33	3.59	0.616	21.865	0	15.814	29.212
2E8Q	265	265	81	21.74	26.32	0.959	1.705	32.8	0.729	3.307
1A05	357	357	24	10.73	11.05	0.59	11.663	4.3	9.941	25.136

2E8S	265	265	77	21.84	25.86	0.961	1.655	34.4	0.77	3.184
1LQO	134	134	60	15.11	17.09	0.621	7.869	0	7.883	15.129
1XSE	274	274	70	19.21	21.67	0.985	1.02	29.3	1.06	2.13
1EQU	284	284	21	7.58	7.77	0.963	1.639	9.1	1.603	3.361
3I3G	143	143	69	18.03	20.39	0.54	7.959	0	8.721	19.465
1V5Z	217	217	23	7.44	7.69	0.579	14.496	0	15.796	28.149
3X22	217	217	52	9.7	11.06	0.59	12.986	3	13.98	25.484
3TY2	245	245	57	3.41	4.17	0.584	13.647	0	11.13	26.812
3BM4	197	197	60	3.53	4.29	0.591	19.674	0	7.361	28.718
4R5M	369	369	29	5.22	5.53	0.58	11.061	4.5	8.494	25.924
1MB4	369	369	29	5.07	5.36	0.659	10.764	2.3	6.349	18.931
1QIN	176	176	46	12.32	12.82	0.54	11.6	0	9.482	23.708
4TTB	189	189	33	6.46	6.79	0.591	14.135	0.9	10.907	25.925

Table S7. The RMSD, TM-score, f_nat, I_RMSD and L_RMSD of DRLComplex for individual targets in the CASP-CAPRI dataset using true contacts and true tertiary structure as inputs. From the table, it can be seen that the RMSDs range from 0.12 to 3.57, with a mean of 0.375 and median of 0.235.The TM-Score, has values ranging from 0.768 to 0.999, with an average of 0.989. For the f_nat metric, values range from 0.96 to 1 with an average of 0.99. I_RMSD has a minimum value of 0.125, a maximum value of 2.979, an average value of 0.326 and a median value of 0.241 and lastly, the L_RMSD has values ranging from 0.261 to 7.106 with an average of 0.8 and median score of 0.533.

Target	RMSD	TM-score	F_nat (%)	I_RMSD	L_RMSD
T0759	0.22	0.9980	100.0	0.224	0.527
T0764	0.18	0.9995	100.0	0.193	0.391
T0770	0.26	0.9992	98.4	0.262	0.562
T0776	0.19	0.9992	100.0	0.200	0.475
T0780	0.15	0.9995	97.8	0.157	0.324
T0792	0.56	0.9851	97.7	0.434	1.393
T0801	0.22	0.9993	99.3	0.256	0.535
T0805	0.16	0.9994	97.4	0.167	0.338
T0811	0.23	0.9991	98.5	0.230	0.516
T0813	0.12	0.9998	98.6	0.125	0.261
T0815	3.57	0.7680	100.0	2.979	7.106
T0819	0.20	0.9994	99.2	0.209	0.424
T0825	0.37	0.9967	100.0	0.326	0.922
T0843	0.24	0.9993	99.4	0.252	0.532

T0847	0.21	0.9989	100.0	0.218	0.468
T0849	0.25	0.9988	99.2	0.264	0.537
T0851	0.22	0.9995	99.0	0.222	0.450
T0852	0.26	0.9991	98.6	0.279	0.636
T0893	0.38	0.9852	99.0	0.278	0.890
T0965	0.38	0.9980	100.0	0.350	0.804
T0966	0.24	0.9994	100.0	0.272	0.633
T0976	0.23	0.9991	100.0	0.209	0.492
T0984	0.29	0.9993	99.1	0.227	0.663
T0999D1	0.38	0.9983	98.1	0.370	0.863
T0999D4	0.28	0.9986	100.0	0.265	0.749
T1003	0.18	0.9996	98.8	0.189	0.393
T1006	0.34	0.9941	96.8	0.347	0.742
T1032	0.20	0.9990	98.2	0.208	0.433
Mean	0.3753	0.9895	99.05	0.2197	0.8235

Table S8. Detailed results (RMSD, TM-score, f_nat, I_RMSD, L_RMSD) of reinforcement learning for CAPS-CAPRI dataset using predicted interchain contacts and true tertiary structure. The TM-score values range from 0.50 to 0.99, with an average value of 0.73.

Target	**RMSD**	**TM-score**	**F_nat** (%)	**I_RMSD**	**L_RMSD**
T0976	18.69	0.54	0	17.8	36
T0776	4.65	0.77	7.6	5.706	13.31
T0813	0.98	0.99	90.1	0.925	2.366
T0852	29.34	0.57	0	21.71	48.98
T0966	27.26	0.53	0	12.55	62.44

T1003	0.89	0.99	91.3	0.833	1.719
T0819	0.62	1	84.3	0.631	1.211
T0965	16.28	0.59	4.3	13.86	28.57
T0792	14.57	0.5	0	14.93	40.23
T0851	1.3	0.98	80.2	0.858	1.819
T0815	10.52	0.51	0	11.9	32.47
T0770	24.99	0.54	0	23.16	50.39
T1032	18.13	0.54	0.9	17.07	29.24
T0999D1	14.42	0.64	0	12.66	27.34
T0805	1.07	0.98	78.4	1.08	2.326
T0780	22.7	0.55	0	21.52	55.84
T1006	16.05	0.5	0	19.34	50.87
T0843	0.92	0.99	91.4	0.942	1.466
T0893	0.85	0.98	94.8	0.92	1.94
T0811	1.08	0.98	86.3	0.901	3.675
T0984	39.94	0.56	1.8	31.43	72.91
T0849	1.07	0.98	71.2	1.086	3.437
T0764	13.7	0.56	3.2	12.97	33.29

T0759	14.09	0.51	0	12.38	38.39
T0999D4	2.48	0.91	71.1	1.661	10.3
T0825	17.39	0.57	7.8	14.91	32.18
T0847	17.08	0.6	0	16.69	55.77
T0801	1.67	0.97	65.24	1.72	3.98
Mean	11.8832	0.73	33.2121	10.4336	26.5163

Table S9. Detailed results (RMSD, TM-score, f_nat, I_RMSD, L_RMSD and TM-score of monomer) of reinforcement learning for CAPS-CAPRI dataset using predicted interchain contacts and predicted tertiary structures. The TM-score values range from 0.36 to 0.97, with an average value of 0.64. The average TM-score of the monomers predicted by Alphafold2 is 0.95.

Target	**TM-score**	**RMSD**	**F_nat (%)**	**I_RMSD**	**L_RMSD**	**TM-score of the monomer**
T1003	0.92	0.58	78	0.51	0.8	0.9964
T0984	0.48	33.57	2	27.17	57.5	0.9805
T0792	0.49	12.7	7	12.16	30.47	0.9585
T0805	0.94	1.83	75	1.85	2.39	0.9675
T0851	0.93	2.03	73	1.68	3.01	0.9687
T0999D1	0.51	16.52	2	13.24	34.22	0.9708
T0815	0.49	14.7	1	11.29	47.88	0.9789

T0759	0.4	15.62	7	12.42	35.6	0.7475
T0893	0.36	20.09	32	15.39	54.24	0.7178
T0825	0.49	18.23	0	15.64	30.46	0.9655
T0819	0.88	1.94	47	2.28	2.22	0.9705
T1006	0.47	11.39	0	11.93	33.96	0.9859
T0966	0.48	32.4	9	30.28	76.42	0.9529
T0770	0.53	11.7	8	13.61	28.94	0.9777
T0843	0.95	1.04	63	0.9	1.72	0.988
T0999D4	0.59	7.03	0	8.09	17.99	0.9767
T0780	0.46	22.17	4	21.1	56.72	0.9599
T0976	0.52	18.16	3	16.99	34.17	0.9777
T0811	0.96	0.88	77	0.87	2.24	0.993
T0852	0.45	33.26	6	21.91	52.15	0.9334
T1032	0.6	6.25	45	5.6	8.49	0.7
T0776	0.75	4.8	26	3.57	18.46	0.9765
T0801	0.88	2.12	57	2.3	4.11	0.9648
T0813	0.97	1.14	69	1.34	1.2	0.9818
T0764	0.55	15.14	1	16.74	27.8	0.9882

T0849	0.88	1.38	64	1.14	1.79	0.9727
T0965	0.57	15.64	1	13.24	27.57	0.9851
T0847	0.52	18.81	2	17.41	50.72	0.9901
Mean	0.64	12.18	27.1	10.74	26.54	0.95

Table S10. Detailed results (RMSD, TM-score, F_nat, I_RMSD and L_RMSD) of DRLComplex on Std_32 with true interchain contacts and true tertiary structures as inputs. The average TM-score is 0.987 with a min of 0.97 and a max of 0.99.

Target	**RMSD**	**TM-score**	**F_nat (%)**	**I_RMSD**	**L_RMSD**
3RRLA_3RRLB	0.93	0.98	98.2	0.902	2.441
2NU9A_2NU9B	0.95	0.99	96.1	0.995	1.678
1EP3A_1EP3B	0.84	0.99	99.3	1.044	1.541
2Y69B_2Y69C	0.74	0.99	84.1	0.756	2.28
3RPFA_3RPFC	0.93	0.97	50	0.864	2.475
1TYGB_1TYGA	0.81	0.99	99.2	0.927	1.986

3MMLA_3MMLB	0.96	0.99	94.8	0.932	2.063
2VPZA_2VPZB	0.91	0.99	93.8	1.172	2.305
2Y69A_2Y69C	0.86	0.99	82.6	0.916	2.123
1I1QA_1I1QB	0.87	0.99	98.8	0.906	2.069
2Y69A_2Y69B	0.75	0.99	86.3	0.842	1.96
1EFPA_1EFPB	0.97	0.99	97.5	1.04	3.152
1W85A_1W85B	0.77	0.99	99.3	0.76	2.263
1ZUNA_1ZUNB	0.93	0.99	98.11	0.954	1.761
3PNLA_3PNLB	0.87	0.99	99.4	0.782	2.077
3OAAH_3OAAG	0.9	0.99	93.8	1.008	1.851
3G5OA_3G5OB	0.86	0.97	90.2	0.839	1.685
2WDQC_2WDQD	0.94	0.97	97.1	0.95	1.853
1BXRA_1BXRB	0.96	0.99	97.6	1.14	3.119
1RM6A_1RM6B	0.79	0.99	96.5	0.604	1.892
1QOPA_1QOPB	0.78	0.99	97.2	0.761	1.625
1B70A_1B70B	0.96	0.99	92.4	1.123	2.165
3A0RA_3A0RB	0.8	0.99	98.1	0.699	2.017
2ONKA_2ONKC	0.9	0.99	96.4	0.535	1.932

2D1PB_2D1PC	0.92	0.97	83.7	0.935	2.325
4HR7A_4HR7B	0.93	0.99	82.6	1.03	2.472
1RM6A_1RM6C	0.95	0.99	74.5	1.323	2.434
3IP4B_3IP4C	0.93	0.99	79.7	0.946	2.628
3IP4A_3IP4C	0.97	0.99	74.3	1.158	2.77
1RM6B_1RM6C	0.81	0.99	69.3	0.9	1.696
Mean	0.88	0.987	90.03	0.92	2.15

Table S11. The table shows the RMSD, TM-Score, f_nat, I_RMSD, and L_RMSD of 31 hetero dimers in Std_32 for predicted contacts. The true monomers are used in this experiment. Also, targets which do not have any interchain contacts are discarded.

Target	**TM-score**	**RMSD**	**F_nat(%)**	**I_RMSD**	**L_RMSD**
1EFPA_1EFPB	0.87	3.19	15	3.24	7.82
1EP3A_1EP3B	0.67	8.08	11	6.45	17.9
1I1QA_1I1QB	0.76	19.29	5	21.2	55.52
1QOPA_1QOPB	0.62	20.38	5	23.72	51.28
1W85A_1W85B	1	0.73	80	0.48	2.02
1ZUNA_1ZUNB	0.65	15.48	8	16.54	30.22
2D1PB_2D1PC	0.49	14.89	4	14.17	34.35

2NU9A_2NU9B	0.89	2.63	48	2.23	5.17
2ONKA_2ONKC	0.52	27.47	2	25.26	65.44
2VPZA_2VPZB	0.84	34.39	2	32.21	92.48
2WDQC_2WDQD	1	0.65	92	0.58	1.12
2Y69A_2Y69B	0.68	26.1	8	27.41	65.18
2Y69A_2Y69C	0.61	25.66	6	25.21	59.85
2Y69B_2Y69C	0.53	20.94	10	16.56	65.58
3A0RA_3A0RB	0.82	17.56	7	20.38	43.67
3G5OA_3G5OB	0.66	8.85	4	9.41	13.83
3IP4A_3IP4B	0.5	25.71	8	20.64	57.94
3IP4A_3IP4C	0.87	12.22	7	14.91	31.5
3IP4B_3IP4C	1	0.67	89	0.44	2.16
3MMLA_3MMLB	0.9	2.64	27	2.36	5.62
3OAAH_3OAAG	0.93	3.79	35	4.22	7.93
3PNLA_3PNLB	0.7	6.95	6	6.29	17.11
3RPFA_3RPFC	0.73	17.47	7	9.07	53.93
3RRLA_3RRLB	0.62	12.91	5	12.63	25.17
4HR7A_4HR7B	0.95	8.35	12	9.6	23.34

1B70A_1B70B	0.82	16.77	2	20.58	34.72
1BXRA_1BXRB	0.81	20.79	6	20.82	49.85
1RM6A_1RM6B	0.74	26.13	6	25.87	59.09
1RM6A_1RM6C	0.74	17.4	9	18.68	48.44
1RM6B_1RM6C	0.65	13.13	4	12.86	29.88
1TYGB_1TYGA	0.91	0.53	79	0.34	0.87
Mean	0.7574	13.9274	19.6451	13.689	34.1606

Table S12. Detailed results (RMSD, TM-score, F_nat, I_RMSD, L_RMSD, TM-score of ligand, and TM-score of receptor) of DRLComplex on Std_32 with predicted interchain contacts and predicted tertiary structures as inputs. The average TM-score is 0.74 with a min of 0.5 and a max of 1.

Target	**RMSD**	**TM-score**	**F_nat**	**I_RMSD**	**L_RMSD**	**TM-score of Ligand**	**TM-score of Receptor**
1EFPA_1EFPB	2.83	0.89	25	3.21	6.08	0.98	0.96
1EP3A_1EP3B	5.07	0.86	16	4.48	11.12	0.97	0.98
1I1QA_1I1QB	20.79	0.67	5	21.52	62.06	0.97	0.97
1QOPA_1QOPB	20.22	0.64	7	23.35	50.21	0.97	0.99
1W85A_1W85B	4.64	0.82	11	3.33	5.88	0.99	0.99
1ZUNA_1ZUNB	16.22	0.67	3	17.45	27.81	0.93	0.95

2D1PB_2D1PC	10.61	0.61	0	10.77	28.47	0.99	0.99
2NU9A_2NU9B	1.92	1	52	1.83	3.41	0.99	0.97
2VPZA_2VPZB	32.16	0.81	1	30.17	89.68	0.98	0.94
2WDQC_2WDQD	6.03	0.75	1	6.07	10.6	0.96	0.98
2Y69A_2Y69B	24.2	0.77	9	23.51	58.93	0.99	0.98
2Y69A_2Y69C	19.97	0.79	1	21.49	40.44	0.99	0.99
2Y69B_2Y69C	18.91	0.56	3	7.11	58	0.98	0.99
3A0RA_3A0RB	21.38	0.6	0	20.33	53.29	0.77	0.90
3G5OA_3G5OB	14.11	0.54	3	14.14	26.31	0.90	0.96
3IP4A_3IP4B	20.31	0.51	4	14.58	81.33	0.99	0.88
3IP4A_3IP4C	11.57	0.82	1	14.83	29.61	0.99	0.92
3IP4B_3IP4C	5.01	0.95	34	2.18	6.99	0.88	0.92
3MMLA_3MMLB	3.83	0.81	40	3.2	9.24	0.99	0.97
3OAAH_3OAAG	14.53	0.81	26	17.55	26.62	0.59	0.97
3PNLA_3PNLB	7.07	0.69	1	6.59	18.67	0.98	0.99
3RPFA_3RPFC	16.36	0.64	10	7.73	54.69	0.97	0.96
3RRLA_3RRLB	7.66	0.73	9	8.03	12.29	0.99	0.93
4HR7A_4HR7B	19.39	0.73	6	20.02	57.84	0.90	0.96

1B70A_1B70B	17.59	0.83	2	21.32	36.1	0.97	0.95
1BXRA_1BXRB	20.93	0.79	2	20.99	50.06	0.99	0.99
1RM6A_1RM6B	26.29	0.6	4	25.94	59.11	0.99	0.99
1RM6A_1RM6C	17.74	0.85	0	18.84	49.08	0.99	0.98
1RM6B_1RM6C	13.9	0.63	5	12.95	29.93	0.99	0.98
1TYGB_1TYGA	0.4	0.97	81	0.41	0.82	0.90	0.96
2ONKA_2ONKC	28.65	0.5	4	25.68	66.83	0.97	0.93
Mean	14.525	0.7367	11.806	13.858	36.177	0.950	0.962

Supplementary Video

To elucidate how reinforcement learning works, we added a video showing the reconstruction of the target 1A2D using true interchain contacts to guide them.

References

1. Dominguez, C., R. Boelens, and A.M. Bonvin, *HADDOCK: a protein− protein docking approach based on biochemical or biophysical information.* Journal of the American Chemical Society, 2003. **125**(7): p. 1731-1737.
2. Chen, R., L. Li, and Z. Weng, *ZDOCK: an initial-stage protein-docking algorithm.* Proteins: Structure, Function, and Bioinformatics, 2003. **52**(1): p. 80-87.
3. Comeau, S.R., et al., *ClusPro: an automated docking and discrimination method for the prediction of protein complexes.* Bioinformatics, 2004. **20**(1): p. 45-50.
4. Gray, J.J., et al., *Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations.* Journal of molecular biology, 2003. **331**(1): p. 281-299.

5. Smith, G.R. and M.J. Sternberg, *Prediction of protein–protein interactions by docking methods.* Current opinion in structural biology, 2002. **12**(1): p. 28-35.
6. Tovchigrechko, A. and I.A. Vakser, *GRAMM-X public web server for protein–protein docking.* Nucleic acids research, 2006. **34**(suppl_2): p. W310-W314.
7. Hwang, H., et al., *Protein–protein docking benchmark version 4.0.* Proteins: Structure, Function, and Bioinformatics, 2010. **78**(15): p. 3111-3114.
8. Wei, X., et al. *EEG-Based Depression Detection with a Synthesis-Based Data Augmentation Strategy*. in *International Symposium on Bioinformatics Research and Applications*. 2021. Springer.
9. Biasini, M., et al., *SWISS-MODEL: modelling protein tertiary and quaternary structure using evolutionary information.* Nucleic acids research, 2014. **42**(W1): p. W252-W258.
10. Mukherjee, S. and Y. Zhang, *Protein-protein complex structure predictions by multimeric threading and template recombination.* Structure, 2011. **19**(7): p. 955-966.
11. Lu, L., H. Lu, and J. Skolnick, *MULTIPROSPECTOR: an algorithm for the prediction of protein–protein interactions by multimeric threading.* Proteins: Structure, Function, and Bioinformatics, 2002. **49**(3): p. 350-364.
12. Baspinar, A., et al., *PRISM: a web server and repository for prediction of protein–protein interactions and modeling their 3D complexes.* Nucleic acids research, 2014. **42**(W1): p. W285-W289.
13. Källberg, M., et al., *Template-based protein structure modeling using the RaptorX web server.* Nature protocols, 2012. **7**(8): p. 1511-1522.
14. Sinha, R., P.J. Kundrotas, and I.A. Vakser, *Docking by structural similarity at protein-protein interfaces.* Proteins: Structure, Function, and Bioinformatics, 2010. **78**(15): p. 3235-3241.
15. Kundrotas, P.J., et al., *Templates are available to model nearly all complexes of structurally characterized proteins.* Proceedings of the National Academy of Sciences, 2012. **109**(24): p. 9438-9441.
16. Negroni, J., R. Mosca, and P. Aloy, *Assessing the applicability of template-based protein docking in the twilight zone.* Structure, 2014. **22**(9): p. 1356-1362.
17. Vakser, I.A., *Low-resolution structural modeling of protein interactome.* Current opinion in structural biology, 2013. **23**(2): p. 198-205.
18. Pierce, B.G., Y. Hourai, and Z. Weng, *Accelerating protein docking in ZDOCK using an advanced 3D convolution library.* PloS one, 2011. **6**(9): p. e24657.
19. Pierce, B. and Z. Weng, *ZRANK: reranking protein docking predictions with an optimized energy function.* Proteins: Structure, Function, and Bioinformatics, 2007. **67**(4): p. 1078-1086.
20. Zacharias, M., *Protein–protein docking with a reduced protein model accounting for side-chain flexibility.* Protein Science, 2003. **12**(6): p. 1271-1282.
21. Gabb, H.A., R.M. Jackson, and M.J. Sternberg, *Modelling protein docking using shape complementarity, electrostatics and biochemical information.* Journal of molecular biology, 1997. **272**(1): p. 106-120.
22. Neveu, E., et al., *PEPSI-Dock: a detailed data-driven protein–protein interaction potential accelerated by polar Fourier correlation.* Bioinformatics, 2016. **32**(17): p. i693-i701.
23. Kastritis, P.L. and A.M. Bonvin, *Are scoring functions in protein− protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark.* Journal of proteome research, 2010. **9**(5): p. 2216-2225.
24. Lensink, M.F., et al., *Prediction of homoprotein and heteroprotein complexes by protein docking and template-based modeling: A CASP-CAPRI experiment.* Proteins: Structure, Function, and Bioinformatics, 2016. **84**: p. 323-348.

25. Janin, J., *Assessing predictions of protein–protein interaction: the CAPRI experiment.* Protein science, 2005. **14**(2): p. 278-283.

26. Mintseris, J. and Z. Weng, *Atomic contact vectors in protein-protein recognition.* Proteins: Structure, Function, and Bioinformatics, 2003. **53**(3): p. 629-639.

27. Wodak, S.J. and R. Méndez, *Prediction of protein–protein interactions: the CAPRI experiment, its evaluation and implications.* Current opinion in structural biology, 2004. **14**(2): p. 242-249.

28. Bradford, J.R. and D.R. Westhead, *Improved prediction of protein–protein binding sites using a support vector machines approach.* Bioinformatics, 2005. **21**(8): p. 1487-1494.

29. De Vries, S.J. and A.M. Bonvin, *Intramolecular surface contacts contain information about protein–protein interface regions.* Bioinformatics, 2006. **22**(17): p. 2094-2098.

30. Chelliah, V., T.L. Blundell, and J. Fernández-Recio, *Efficient restraints for protein–protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment.* Journal of molecular biology, 2006. **357**(5): p. 1669-1682.

31. Senior, A.W., et al., *Improved protein structure prediction using potentials from deep learning.* Nature, 2020. **577**(7792): p. 706-710.

32. Yang, J., et al., *Improved protein structure prediction using predicted interresidue orientations.* Proceedings of the National Academy of Sciences, 2020. **117**(3): p. 1496-1503.

33. Adhikari, B., et al., *CONFOLD: residue-residue contact-guided ab initio protein folding.* Proteins: Structure, Function, and Bioinformatics, 2015. **83**(8): p. 1436-1449.

34. Brunger, A.T., *Version 1.2 of the Crystallography and NMR system.* Nature protocols, 2007. **2**(11): p. 2728-2733.

35. Eswar, N., et al., *Comparative protein structure modeling using Modeller.* Current protocols in bioinformatics, 2006. **15**(1): p. 5.6. 1-5.6. 30.

36. Ovchinnikov, S., et al., *Protein structure prediction using Rosetta in CASP12.* Proteins: Structure, Function, and Bioinformatics, 2018. **86**: p. 113-121.

37. Quadir, F., et al., *DNCON2_Inter: predicting interchain contacts for homodimeric and homomultimeric protein complexes using multiple sequence alignments of monomers and deep learning.* Scientific reports, 2021. **11**(1): p. 1-10.

38. Zhou, T.-m., S. Wang, and J. Xu, *Deep learning reveals many more inter-protein residue-residue contacts than direct coupling analysis.* BioRxiv, 2018: p. 240754.

39. Soltanikazemi, E., et al., *Distance-based Reconstruction of Protein Quaternary Structures from Inter-Chain Contacts.* bioRxiv, 2021: p. 2021.05.24.445503.

40. Ganea, O.-E., et al., *Independent se (3)-equivariant models for end-to-end rigid protein docking.* arXiv preprint arXiv:2111.07786, 2021.

41. Roy, R.S., et al., *A deep dilated convolutional residual network for predicting interchain contacts of protein homodimers.* Bioinformatics, 2022. **38**(7): p. 1904-1910.

42. Xie, Z. and J. Xu, *Deep graph learning of inter-protein contacts.* Bioinformatics, 2022. **38**(4): p. 947-953.

43. Jumper, J., et al., *Highly accurate protein structure prediction with AlphaFold.* Nature, 2021. **596**(7873): p. 583-589.