Leipzig University
Fakultät für Mathematik und Informatik
Computer Science, B.Sc.

Extending the capabilities of AI structure prediction tools to reconstruct the locations of water molecules

Bachelor's Thesis

Maximilian Kleineberg
Born Oct. 22, 2002 in Essen

Matriculation Number 3773001

1. Referee: Prof. Dr. Jens Meiler

Submission date: September 23, 2025

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work. AI tools like Google's Gemini or ChatGPT were used to create code blueprints for some of the scripts mentioned in the section Data availability.

Leipzig, September 23, 2025

Maximilian Kleinberg

Abstract

The field of protein structure prediction has had several breakthroughs in the past decade. State-of-the-art tools like AlphaFold 3 offer capabilities to not only predict protein structures but also non-proteinogenic biomolecules associated with proteins like ligands or metal ions. Still, a small molecule with major importance for protein stability and function, has been overlooked in these: water.
This thesis aims to address this gap. I present a finetuned version of the Boltz structure prediction tool, trained on data containing water molecules. First evaluations show a small improvement of the finetuned model in its ability to predict water positions over the vanilla Boltz model. This shows the potential for integrating water molecules into structure prediction models to achieve a more complete view of biomolecular systems and opens the space for more research. Here, I will present the processes of curating the dataset, training, and evaluation of the model.

Contents

Chapter 1

Introduction

Over the past decade, machine learning has led to breakthroughs in many areas, such as computer vision [1], natural language processing [2] and weather forecasting [3]. Deep neural networks have demonstrated an unprecedented performance across different scientific fields. One of the fields revolutionized by generative models, was structural biology, and in particular the so-called protein structure prediction problem.

The structure prediction problem has been a major focus of bioinformatic research in the past three decades [4]. The ultimate goal is to predict a protein's 3D structure given an amino acid sequence encoding it [5]. Even though a protein 3D structure can be determined experimentally, this process is very labor-intensive and thus expensive. For this reason, development of computational tools that can approximate the experimental structures is desirable and has been a major focus of research. This is an incredibly hard problem, as the space of theoretically possible 3D conformations even of a small protein given its sequence, is large. To this day, this problem has not been fully solved.

Historically, research in protein structure prediction has been centered around physics-inspired energy functions. These functions, like Rosetta software [6] [7], were designed to predict the energy state of a 3D atomic configuration of a protein. Computers then minimized these energy functions in order to create a 3D model [8]. Although classical modeling tools led to highly accurate description of many protein systems, their energy functions are parameterized by empirical experimental observations and are reliant on some unrealistic approximations. Recently, these methods were surpassed by deep generative models like AlphaFold2 that demonstrated an unprecedented, near-Angstroem modeling accuracy of protein structures [9].

Although AlphaFold2 has set a new standard for accuracy in structural biology, it is limited to modeling single protein chains or protein–protein interactions. This oversimplifies our understanding of protein structure and function,

because it treats proteins in isolation, without considering their biologically relevant non-protein partners. To address this inherent limitation, the very recently released AlphaFold3 [10] and Boltz [11] models have revisited their architectures and added capabilities to model proteins with non-proteinogenic counterparts, such as ligands, metal ions, and nucleic acids. (For more information on AlphaFold 2 and 3 and Boltz see 2.4 and 2.5.) However, an essential molecule has been completely overlooked by all state-of-the art models. This molecule is water.

In most realistic applications, biomolecules are always dissolved in water or, in other words, surrounded by water molecules. These waters are key to protein folding, protein stability, and function [12], and without modeling them one cannot fully understand protein function. A common example of proteins that depend on water molecules for their function are hydrolases. In glycosyl hydrolases for instance, water is being used as a reactant for the hydrolysis of a glycosidic bond [13]. One more example would be the structure of bovine rhodopsin (PDB entry 1L9H) that was researched in [14]. Among others, the paper outlines three water molecule in the protein's transmembrane region. By forming a hydrogen-bonded network, these waters mediate the interaction of multiple different chains of the protein. The disruption of this network is likely involved in the protein activation process.

Modern structure prediction tools often model water implicitly, assuming the biomolecule is solved, however to better understand (and design) proteins and their interactions, it is sometimes beneficial to explicitly know certain water molecule's positions [15]. Software for water molecule location prediction has been developed, such as GalaxyWaterCNN [15], which takes a biomolecule 3D-structure and populates it with waters. A disadvantage of this approach is that often water molecules influence the biomolecule's structure. Therefore knowing the water positions would necessitate a reevaluation of the biomolecule structure in turn necessitating a reevaluation of the water positions.

With current tools, it is possible to add ligands like water to a structure prediction model. My aim is to create a model that can predict the 3D-structure of biomolecules and, in the process, also predict the positions of water molecules. To achieve this, I first created a dataset of biomolecule coordinate files from PDB that is filtered and made suitable for training a neural network on structure prediction with water molecules. I explain how I do this in detail in chapter 3. Then I finetuned Boltz [11] using this data. By finetuning it, I aim to extend its capabilities to also predict water molecule positions. In this work, I will refer to the baseline model as "Boltz-1" and the finetuned model as "finetuned-Boltz". Finally, I evaluated my model in comparison to the pretrained Boltz model.

Chapter 2

Background

This chapter provides a deeper review of relevant literature and concepts necessary to understand the problem and the process of finetuning the Boltz model.

2.1 Protein Data Bank - PDB

The Protein Data Bank (PDB) [16] is a publicly accessible data archive that contains 3D information on a large set of different biomolecules. In addition to proteins it stores 3D data about nucleic acids, ribosomes and even entire viruses. The data is saved and made accessible in coordinate files and gathered by laboratories all around the world [17]. As of the date of 2025-08-28, there were 241,345 structures available in the PDB archive. [[18] under https://www.rcsb.org/stats/growth/growth-released-structures] Each PDB entry has a unique associated PDB ID with which a structure can be easily identified.
A coordinate file contains a multitude of information. First of all, it contains metadata about information like the entry's ID and author(s) of the study that researched the respective biomolecule. Then, if the file contains proteins or nucleic acids, information about the sequences is given. There is also often a description of the different parts of a larger biomolecule, for example for proteins the individual amino acids are listed together with data like their torsion angles. Finally, the main part of a coordinate file is a listing of all atoms together with their coordinates and additional atom-specific data like B-factor (see 3.2.4). With this rather flexible structure, many different macromolecules can be saved in the same format. [17]

2.2 Structure determination methods

In order to gather 3D data of biomolecules, several methods can be employed. The most important ones are X-ray crystallography, NMR spectroscopy, and electron microscopy. It is noteworthy that in each of these approaches the final model is a mixture of experimental observation and knowledge-based modeling. I will now briefly summarize the 3 aforementioned methods. [17]
The majority of PDB structures are determined using X-ray crystallography. For this, the protein is purified into a crystal structure which is then subjected to an X-ray beam. The X-rays diffract around the protein leading to an X-ray pattern that can be measured. This pattern can be used to determine electron density in different locations of the protein. This can then, together with additional data like the protein's amino acid sequence (which is often known in advance), be used to reconstruct the atom locations within the protein. This approach works well for proteins that form crystal structures easily but falls apart for proteins that do not form crystals. [17]
In Nuclear Magnetic Resonance (NMR) spectroscopy, the protein is put into a solution, exposed to a strong magnetic field and probed by radio waves of different frequencies. The radio waves excite the protein atoms that then resonate upon returning to a non-excited state. Surrounding atoms also determine how high or low these frequencies are. These resonance frequencies are then recorded. Computer simulations help to reconstruct an atom model from these frequencies. This method has the advantage of not needing the protein to be in crystal form, however so far it only works for small and medium-sized proteins. [17]
In electron microscopy a protein sample is prepared and placed into an electron microscope. A beam of electrons is used to image the protein directly. The result are multiple 2D images of the protein in different orientations. These images can then be combined to reconstruct the 3D structure of the protein. Electron microscopy as a 3D structure solving tool is a rather new technique that is increasingly gaining popularity. However, it still struggles to achieve high resolutions (a measure for a structure's quality, more on that in 3.2.2) like other methods can. This, however, might be subject to change in the coming years. [17] [19] [1]
PDB will be our basic data source for this work. The challenge will be to screen for the right structures and to process them in a way that helps to train the Neural Network. I describe this in detail in section 3.

[1] Similar, or even the same methods can also be employed to determine the structures of other biomolecules, such as nucleic acids. For this, see the website of Uni Ulm: https://www.uni-ulm.de/nawi/institute-of-pharmaceutical-biotechnology/institute/x-ray-crystallography-facility/

2.3 Multiple sequence alignment (MSA)

As proteins are part of living organisms they too underlie evolution. Closely related species often have closely related, and thus similar, proteins. Protein mutations are usually conservative, meaning that the overall structure and function of a protein are typically conserved over evolutionary time even if the specific amino acids change. There can be distantly related proteins that differ by 80% in amino acid sequence yet they are almost identical in structure [20]. This information can be used to better predict protein structure as related proteins often have related structures.
Multiple sequence alignment (MSA) arranges protein sequences into a table where residues (like amino acids), that are in the same column, are homologous. That means that these protein residues are either derived from a single position in an ancestral protein, superposable, or play a common role in the protein's function. [21]
When residues in a protein are spatially close to each other even if they are far apart in the sequence, they always tend to co-evolve. This means that given data on co-evolution, one can estimate the physical proximity of residues which is crucial for modern structure prediction tools. This co-evolutionary data can be gained from examining MSA data [22]. Exploiting this data has led to major advances in structure prediction [9].
In this work, I also use MSA data for both training and prediction. In both cases I use the MSA data provided by the Boltz authors [11].

2.4 AlphaFold

In order to develop structure prediction tools, there is the Critical Assessment of Structure Prediction (CASP) competition. In this competition, that happens every two years, multiple structure prediction tools compete in several categories to perform the best structure predictions on yet unpublished and thus unseen structures. The competition has made it possible to benchmark and compare state-of-the-art structure prediction tools. [23] [24]
In the CASP 14 competition in 2020, the deep neural network-based model called Alpha Fold 2 achieved impressive results revolutionizing protein structure prediction. Alpha Fold 2 has been the first computational method to provide predictions with near experimental rates of accuracy [9] on an unbiased set of proteins unseen by the model during training.
This was achieved by adapting a new transformer architecture, a concept that was originally proposed for natural language processing. Also, evolutionary data was heavily used for predictions (see 2.3). Furthermore, Alpha Fold 2 has

used a recycling mechanism that processes the previously made predictions, making prediction iterative. This yields better results than one-shot prediction. Finally, another factor of Alpha Fold 2's success has been the ever growing training dataset as PDB has been constantly growing up until the development of Alpha Fold 2 and also continues to grow to this day. [20]
Alpha Fold 2 was widely adopted even shortly after its introduction [20] in 2020. This led to experimentation with and an interest in further development of the tool. It was shown that simple input modifications on Alpha Fold 2 would yield surprisingly good results for protein interaction predictions. On the basis of this, Alpha Fold 3 was developed. The main advancement of Alpha Fold 3 was its ability to not only predict protein structures and protein-protein complexes but also complexes with non-proteinogenic molecules, including nucleic acids, ions, and ligands. In this way it was possible to accurately model the biological context of proteins where several associated non-protein molecules significantly impact the protein's biological function. Alpha Fold 3 reaches high accuracy on protein-ligand interactions and protein-nucleic acid interactions. [10]
This was achieved by a few new innovations, the most notable of which were a diffusion-based architecture, which I will explain in 2.5.1, an improved training procedure, and an all-atom biomolecule representation that contrasts with AlphaFold2's coarse-grained backbone representation. [10] [9]

2.5 Boltz-1

Boltz-1 was published in 2024 by [11]. It is another structure prediction tool that is very similar to Alpha Fold 3 in approach, architecture, and results. That means it also explicitly can do structure prediction on a multitude of different biomolecules. A major difference from Alpha Fold 3, however, is that Boltz-1, meaning its training and inference code, model weights, corresponding datasets, and benchmarks, is completely open-source under the MIT license. This open-source nature makes it highly accessible for research and modification. This was done in order to "foster global collaboration, accelerate discoveries, and provide a robust platform for advancing biomolecular modeling"[11]. For this reason and because it is made to also explicitly predict ligand locations, I use Boltz as the main tool in this work by finetuning it to increase its capabilities to also predict water molecule locations.
Boltz-1 includes several rather new features and innovations. For starters, it uses a diffusion architecture which I will explain further in 2.5.1 that has been improved from the Alpha Fold 3 architecture. Also, Boltz-1 includes improvements in speed optimization and a revised confidence model. Furthermore,

there have been innovations in data processing which I will discuss in 2.5.2. Lastly, Boltz-1 uses a method called Boltz steering which will be explained in 2.5.3.[11]

2.5.1 Diffusion architecture

Diffusion models were originally created for image generation, however they now play an important role in many applications, including structure prediction, for which they have become an invaluable tool. [25]
In diffusion, there are two processes: forward diffusion and reverse diffusion. Both can be modeled as Markov chains. The first Markov chain $x_0, x_1, ..., x_T$ turns data (represented as the vector x_0) into noise(x_T). This process is typically hand-designed with the aim to turn the data iteratively into a random distribution. For this, we define a kernel $q(x_t|x_{t-1})$. With this $q(x_1, ..., x_T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$. A common choice for the target distribution is the Gaussian distribution and a common kernel for that is $q(x_t|x_{t-1}) = \mathcal{N}(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$ with β being a hyperparameter in $(0, 1)$. If the kernel is applied repeatedly the data loses its structure and slowly turns into noise, distributed according to, in this case, the Gaussian distribution.[26]
The reverse diffusion process in turn constructs data from a random distribution. For this, reverse diffusion also uses a transition kernel. This kernel however, consists of learnable parameters. Training these is the crucial step. In the end, the trained parameters should yield a kernel $p_\theta(x_{t-1}|x_t)$ that helps form the Markov chain $x_t, ..., x_0$. Because we want to reverse the forward diffusion process, the reverse probabilities should closely match the forward probabilities as in $p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \approx q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1})$. This can be achieved by minimizing the Kullback-Leibler divergence between the two by manipulating the learnable parameters θ. [26]
For generating new data with the network, a vector x_T is taken that follows our distribution but is otherwise completely random (pure noise). Now we can apply only the reverse diffusion process that we learned before, yielding a Markov chain $x_T, x_{T-1}, ..., x_0$. Our newly generated data is then the vector x_0. It is important to understand that this process is non-deterministic and the final result heavily depends on the initial randomly sampled starting vector x_T. [26]

2.5.2 New dataset curation techniques

Most proteins are sequenced individually, so there is MSA data only on singular proteins. That makes it hard to predict protein-protein interactions. However,

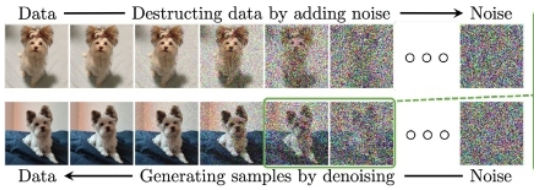


Figure 2.1: This image from [26] shows the forward (top) and reverse (bottom) diffusion processes, in this case for its original use, image generation.

it is possible to approximate co-evolving protein pairs using taxonomic data. [11] provide a MSA pairing algorithm to gather this MSA pairing data.
As biomolecule complexes can get very big, it can get impossible to fit them into the VRAM of even very powerful GPUs. This is why it is necessary to crop structures. There are two important traditional approaches to do that: The contiguous approach chooses crops to be consecutive residues in some sequence. The spatial approach crops tokens simply as a function of distance to a center token. [11] interpolate between these approaches. This means neighborhoods consisting of contiguous portions of a sequence around a specific token are defined. These neighborhoods then get added to the crop depending on their distance to the crop's center token. The neighborhood size is a central consideration here; if it is 0 the method turns into spatial cropping while the method turns into contiguous cropping as the neighborhood size increases to half the size of the crop's maximum token budget. In Boltz-1 the neighborhood size is randomly sampled uniformly between 0 and 40 tokens for every training sample.
While AlphaFold 3 provides the option to add information about a protein's binding pocket to improve prediction, this requires knowing all pocket residues, which is often unrealistic. Additionally, it necessitates maintaining two separate models — one with pocket-conditioning and one without. To address these problems, [27] developed a new, robust pocket-conditioning algorithm. During training, in 30% of iterations, a binder is randomly selected. Intentionally incomplete pocket information—where only some residues are randomly selected from the pocket—is then added to the training case.

2.5.3 Boltz steering

The authors of Boltz-1 [11] discovered that their model sometimes had instances of hallucination in its output. Most prominently, often times entire chains were placed on top of each other. Several other instances of unphysical predicted structures were found with problems like steric clashes between atoms, incorrect bond lengths and angles, incorrect stereochemistry, and aro-

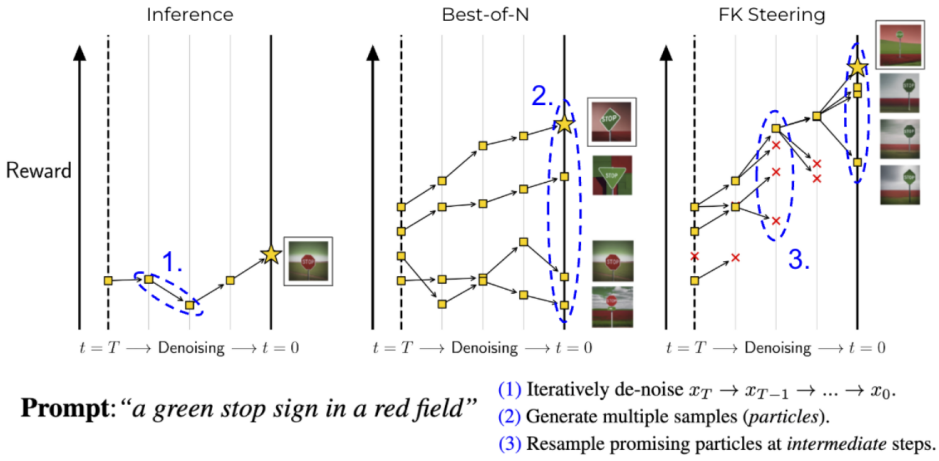


Figure 2.2: This image from [28] gives an intuitive overview of the steering process. Only the "most promising" process samples are used for further inference.

matic rings that were predicted to not be planar. This can prevent the predictions from being used in applications like drug design. In order to solve these problems, Boltz steering was released on 2025-04-26.
In Boltz, the steering process follows the one introduced by [28]. Steering is active during the diffusion inference. Different diffusion processes are sampled at intermediate steps during the diffusion process. A scoring function is used to evaluate these samples. Only sampled processes that have a high scoring value will be used to continue the diffusion inference process. In this way one can "steer" the diffusion process towards reasonable results. Figure 2.2 gives an intuitive overview of the idea.
Boltz adapts this approach. The scoring function in this case is a function that evaluates the physical correctness of a structure. To be precise, a low energy which indicates a physically plausible structure corresponds with a high value of the scoring function. In this way, the diffusion process is nudged towards physically more plausible structures. For this, Boltz's scoring function takes into consideration atom chirality, bond stereochemistry, planarity, internal geometry, steric clashes, overlapping chains and covalently bonded chains. [11]
For inference, I too use Boltz steering to achieve overall better results. However the scoring function remains as it is in the original Boltz model.

2.6 Coordinate file manipulation with ProDy

ProDy is a python package to work with coordinate files like .pdb or .cif. In this way it can work with both experimental data and theoretical data like predicted structures. Crucially, this also allows me to compare experimental and predicted data. [29]
ProDy's basic data structure is an atom group, which contains a collection of atoms and their corresponding data, most notably their spatial coordinates and other attributes like their Beta value. With ProDy it is also possible to add new properties to atom groups, like I did with the Z-value (explained in 3.2.4). Another key function is the ability to select subsets of atoms using selection strings. This allows me to filter atom groups in a very detailed fashion which will be important for dataset curation. [[29] and corresponding tutorial: http://www.bahargroup.org/prody/tutorials/prody_tutorial/]
Another important function is the possibility to calculate several structural properties like singular atoms' radius of gyration. Also, ProDy can be used to align multiple structures. This is especially important for comparing structures. For example it can be used to calculate the rmsd value between two structures[29]. This was also used for evaluation in this work, see 4.4.

Chapter 3

Dataset curation

3.1 Data source

The data for finetuning Boltz was entirely taken from the RCSB PDB dataset mentioned in 2.1. In order to ensure comparability between the Boltz-1 model and finetuned-Boltz model, I only used PDB data that was also used for training Boltz-1. That means that I only considered PDB files released before 2021-09-30. Note that [11] also used the OpenFold distillation dataset which was not used here.
Then I filtered and preprocessed this PDB data in order to make it suitable for water molecule position prediction. This process will be outlined in 3.2. Finally, the data was preprocessed further using the data pipeline used and provided by [11].

3.2 Data selection and filtering

A central challenge was to filter the data to make a finetuning possible. I want to consider only waters about whose exact position I can be fairly sure. For this reason, I need to exclude water molecules from the data whose position can vary a lot as it would be impossible to predict their positions thus only adding noise to the dataset.
As biomolecules are typically embedded in a hydration shell, [30] there are some waters whose exact positions are more important for the biomolecule's structure and function than others. As an example, take the bovine rhodopsin from chapter 1. There, three waters in the transmembrane region are outlined that have a special importance for the protein's structure and function. Here, I will aim to keep waters like these but filter out waters in the wider outer hydration shell that do not contribute as directly to the protein's structure

and function.
In this section I will go through several factors and decisions I made upon these factors to prepare my dataset.

3.2.1 Existence of water data

Usually, PDB files contain at least some water molecules. Some PDB files however, do not contain any waters like the entry with id 2ft9. These files were filtered out because they do not help us to predict water molecule positions and would just unnecessarily decelerate training.

3.2.2 Resolution

Resolution is a file-wide measure to assess the overall structure quality of a PDB file. It measures how well adjacent atoms in a structure can be distinguished. [1]
[11] used all eligible PDB structures with a resolution ≤ 9.0 Å. For my task, I do not need that many training files, especially because waters are very common molecules. However, I want to be fairly certain about a water's position within the biomolecule. Therefore, I decided for the stricter 2.5 Å resolution restriction to leverage sample quality over number of samples.

3.2.3 Occupancy

As described in 2.2, structures are often brought into a crystal form before being examined and subsequently uploaded to the PDB. Sometimes, there are slight differences between the different units that make up the crystal. For example, a sidechain can have different conformations. The Occupancy value of an atom is the fraction of crystal units where the atom has the specified coordinates. [2]
For my training I only want to include waters of whose position one can be sure in every conformation. For this reason I only include waters with an occupancy of 1.0.

3.2.4 B-factors

B-factor, also called B-value, β-value, or temperature factor, is an indirect measure of uncertainty about an atom's exact position. In this way it is similar

[1] [18] (the information about resolution is not in the article itself but can be found under `https://www.rcsb.org/docs/general-help/assessing-the-quality-of-3d-structures`)
[2] [17] - information under `https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/`

to resolution but on an atom-wide (not file-wide) level. The uncertainty can originate from vibrations of the individual atoms or slight differences in the positions of atoms within different molecules in a crystal lattice. These effects lead to a smearing of the electron density which is measured to identify atom positions. High B-factors signify high uncertainty. Typically the B-factor of atoms on the outside of a large biomolecule like a protein are higher, as atoms there move more while B-factors of atoms deep within a protein tend to be lower. [17] [31]
B-factor distribution can vary significantly between different PDB files. This is often not an attribute of the structure studied but of the methodology. For starters, fundamentally different methods are used to examine molecule structures like X-ray crystallography and NMR spectroscopy (see also 2.2). [17] Also, there can be noticeable variation in B-factor distribution even when using the same method like X-ray crystallography as mentioned in [32]: The source of variation can be both inherent to X-ray experiments or produced by the computational processing of the data. This means both factors such as beam alignment as well as factors like signal-to-noise cutoff among many more can heavily impact the distribution of B-factors. Even when examining different crystals of the same structure, different B-factor distributions can be found between individual experiments.
This means while looking at B-factors within a single file gives us a lot of information, comparing B-factors directly between different files makes little sense. However, I still want to work with multiple PDB files. To solve this, B-factors can be normalized over the distribution of B-factors within each file. These normalized values are then comparable even between files.
In Figure 3.1 we can see that B-factors are approximately normally distributed over all water molecule oxygens across multiple files when we normalize only over water molecules in each file (with the formula $B'_i = \frac{B_i - \overline{B}}{\sigma(B)}$). When we normalize over the B-factors of all atoms however, the distribution is skewed towards higher normalized B-factors. This means that the position of a water molecule is on average less certain than the position of an average atom. This is because while most atoms in a PDB file belong to a larger biomolecule like a protein which is relatively rigid, water molecules are much more free to move around. Additionally, water molecules tend to be in the outer layers of a biomolecule where, as mentioned above, B-factors are typically higher.
For our purposes the distribution of B-factors over all atoms in a PDB file is more interesting. On this basis I calculate the Z-value for each water oxygen. The Z-value of an atom is the B-factor of said atom normalized over the B-factors of the entire file. That means $Z_i = \frac{B_i - \overline{B}}{\sigma(B)}$ with sigma(B) being the standard derivation of the B-factor over all atoms of the file.
Water oxygens with low Z-values are atoms about whose position we are rea-

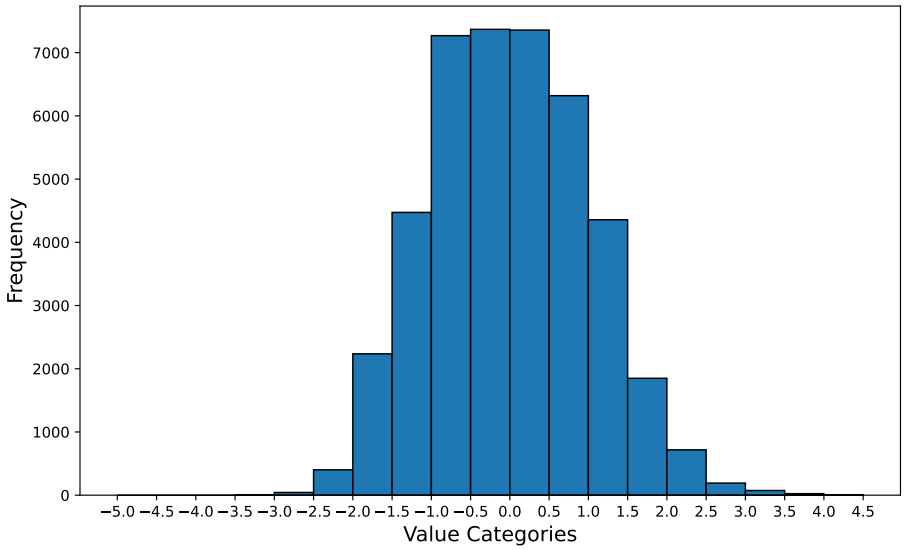
sonably certain regardless of the fact that they belong to waters. Thus, I will go on to filter out waters with a Z-value of 0.5 and above, keeping waters with comparatively low B-factors.

3.2.5 Hydrogen bonds

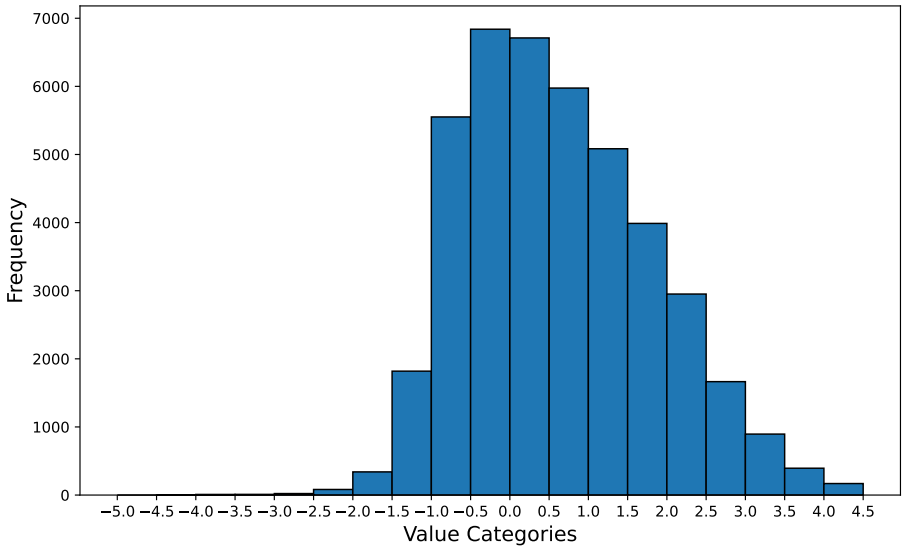
Hydrogen bonds between biomolecule atoms and water atoms can be crucial for both biomolecule structure and function as we have seen before in [14]. Additionally, hydrogen bonds keep water molecules "in place", making their positions more certain. We can see that this assumption is generally true in Figure 3.3. The more hydrogen bonds a water molecule has, the lower the Z-value of its oxygen atom, indicating higher confidence in the atom's and, by extension, the water molecule's position.
However finding hydrogen bonds from PDB data turns out to be challenging. Prody (see 2.6) offers functionality to calculate hydrogen bonds but to do this it requires data on the positions of hydrogen atoms. While some PDB files contain hydrogen atoms, the vast majority come without hydrogen atom data. There are tools to predict hydrogen atom positions like pdbfixer [3] but I found that they do not work well enough for my purposes and introduce unreasonable amounts of bias.
To circumvent this issue I chose a much simpler approximation of hydrogen bonds. For my purposes a water-biomolecule hydrogen bond is a water in close proximity (3.5Å or closer) to a polar atom (in our case S, O or N) belonging to anything that is not a water, including proteins, nucleic acids and cofactors. For simplicity, bond angles were not considered here. See the distribution of number of water molecules in Figure 3.2.
Here it is important to mention that this is by far no perfect approach. In Figure 3.2 we can see that some waters allegedly have 7 or 8 hydrogen bonds which seems chemically unrealistic. Nevertheless these cases are rare enough that I deemed them to be not relevant enough for now.
Because this work mainly focuses on water molecules that have some relevance for biomolecule structure and function I can be quite restrictive with water molecule filtering with regards to the number of hydrogen bonds. I filter out all waters that have fewer than 2 hydrogen bonds.
Sometimes, PDB files contain metal ions. These usually serve important functions in biomolecule structure and function. What is interesting here is that they often interact with water molecules [33] [34] rendering these water molecules important for my purposes. For that reason I also accept waters that have a water - metal ion hydrogen bond even if they only have one hydrogen bond in total. The other filters concerning occupancy and B-factors

[3] `https://github.com/openmm/pdbfixer`

are however kept in either case. A water - metal ion bond is considered to be a water molecule in close proximity (3.0Å or smaller) to a metal ion. The metals Mn, Fe, Co, Ni, Cu, Zn, Ca, Na, K, and Mg are considered. 3.0Å was a better proximity indicator than 3.5Å for these kinds of bonds because otherwise too many unrelated water atoms would be kept in the data because of their proximity to a metal ion.



(a) Distribution of B-values of water oxygens normalized file-by-file over all water oxygens of the file



(b) Distribution of B-values of water oxygens normalized file-by-file over all atoms of the file

Figure 3.1: Distribution of B-values of water molecule oxygens with different normalizations within a sample of PDB files

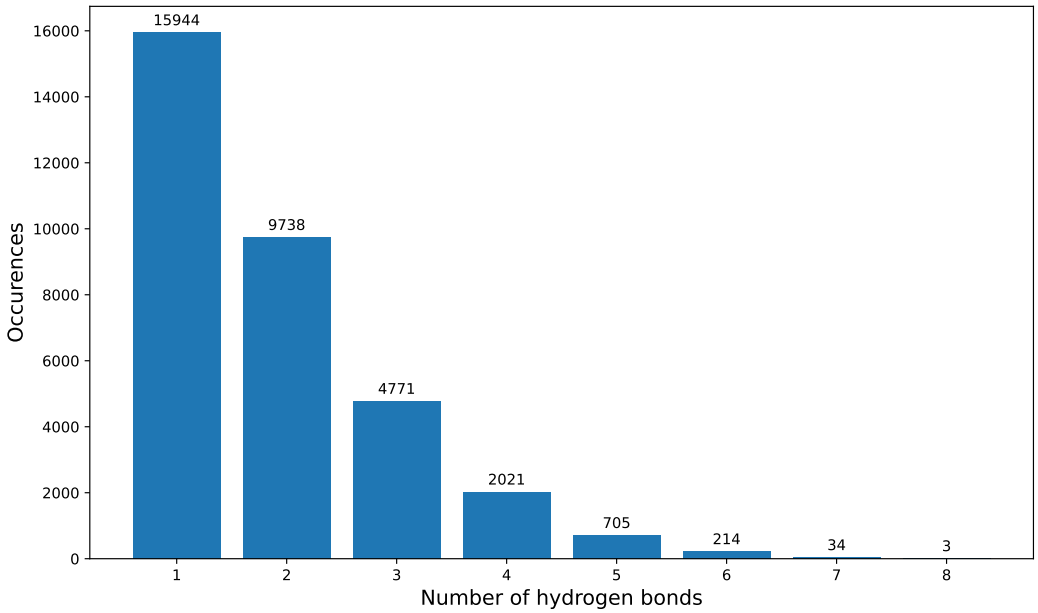


Figure 3.2: Distribution of the number of water-biomolecule hydrogen bonds (as by my definition) per water in the PDB sample from before.

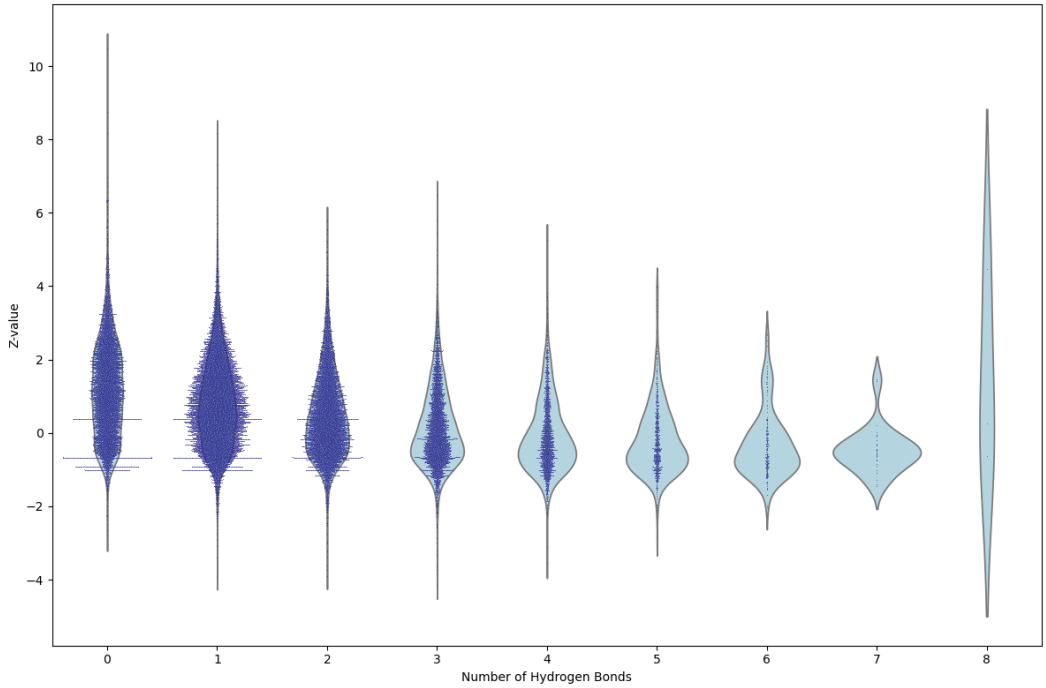


Figure 3.3: Z-value of a water oxygen as a function of the number of hydrogen bonds the water has (according to my H-bond definition). In general we can say the higher the number of hydrogen bonds the lower the Z-value meaning lower uncertainty. The PDB files are from the same sample as before.

Chapter 4

Results

4.1 Dataset curation summary

One major challenge to finetune the model, was to curate a suitable dataset. The process is explained in detail in chapter 3. Here is a summary of the steps I took to curate the dataset. First, I take all PDB files provided by [11] (216,870 structures). These are all PDB structures released before 2021-09-30 and with a resolution of at least 9.0Å. Then I applied the following additional filters:

File-wide filters:

1. Exclude files containing no waters
2. Exclude files with a resolution of 2.5 or higher

This leaves 143,131 structures. The validation data was filtered in the exact same way leaving 409 validation structures.

Atom-wide filters

1. Filter out all waters that have an occupancy value below 1.0
2. Filter out all waters that have a Z-value of or above 0.5
3. Filter out all waters that have fewer than 2 hydrogen bonds to non-waters
4. Add back in all waters that have one metal-water hydrogen bond that were rejected by point 3

Finally, I ran the data processed in this way through the Boltz data preparation pipeline as provided by [11]. Note that resources like MSA data and the CCD dictionary were taken directly from [11] without customly preprocessing them.

4.2 Training

The training was initiated from the Boltz-1 pretrained checkpoint available on Hugging Face [1] and done entirely with my customly preprocessed PDB data (see chapter 3) from that point on. For validation the same split as in Boltz-1, was used. I used a maximum number of tokens of 256 and maximum number of atoms of 2304. Furthermore I used 10,000 samples per epoch for training. This is far different than what was used for training Boltz-1 where 100,000 samples per epoch were used, according to the default training configurations in the Boltz repository. The reason for that is that I did not have access to hardware that is as strong as the hardware used by the Boltz authors. Still choosing 100,000 samples per epoch would have yielded a very long training time per epoch and thus a long time until validation could be done as validation happens at the end of every epoch. Also my task was not to train the model from scratch but to finetune it so not as many samples are needed.
Other than that, I used the configurations from the Boltz repository in the file structure.yaml. [2] That means, I accumulated gradient over 128 batches with a batch size of 1. For the final configuration file, see section Data availability. Also, I did not train the confidence model. This however was an error, that I realized only after the training was done.
In total, I trained until step 1422 in epoch 17, having accomplished a total of 171,422 steps.

4.3 Prediction queries

To make a Boltz prediction, a query is required. This can be either in .yaml or .fasta format. Here I will focus on the .yaml format, which is also the recommended format.
A .yaml query file consists of up to four sections. The most important section, sequences, has one entry for each unique chain or molecule. For proteins and nucleic acids, the corresponding sequence (amino acid sequence/ nucleotide sequence) needs to be provided. For ligands in contrast, either a SMILES string or a chemical component dictionary (CCD) code has to be provided. Also, each sequence needs a unique ID. For proteins, there is the possibility to add MSA data. (see 2.3) The constraints section allows to give additional information about the structure. In the template section a template to base the prediction on can be added. This is another atomic coordinate file (.pdb or .cif) on which the prediction should be based. Finally, the properties section

[1] `https://huggingface.co/boltz-community/boltz-1`
[2] `https://github.com/jwohlwend/boltz/blob/main/scripts/train/configs/structure.yaml`

can be used to calculate the resulting protein's affinity to a given small molecule ligand. [3]
For my work, I use queries that have been given as examples by the Boltz's authors to test Boltz-1 [4] and modify them to include water molecules as ligands by adding n water molecules to the sequences section. With this approach, it is possible to make predictions with both Boltz-1 and modified-Boltz. We will see in section 4.4 how the predictions compare.

4.4 Evaluation

In the original paper [11], Boltz has been validated and compared using two different datasets, the one from the CASP15 competition (66 datapoints) and an internal validation dataset (internally called "test") that contains 542 files. For my evaluation I use the internal dataset, because it contains more files. I then preprocessed this dataset in the same way as I preprocessed the training dataset (see 4.1) in order to only validate on files and waters that were deemed "relevant" before. This formed the evaluation dataset.
The performance of the finetuned "modified-Boltz", will be compared to the performance of Boltz-1 that uses the model from the initial checkpoint from Huggingface. [5]
Because the number of water molecules that should be predicted has to be set in the query, I made two sets of predictions for both models: one set of predictions where the queries contain a fixed number of 30 water molecules and one where the queries contain as many water molecules as the ground truth file has. In this way, it is possible to compare how well modified-Boltz performs on different numbers of water molecules.
For predicting the evaluation samples, I used 3 recycling steps and 200 sampling steps for both models. The step size for inference was 1.638 . Overall, all prediction parameters were the default parameters for Boltz. The confidence model was turned off for inference, because it was not trained.
For evaluation, the predictions were first superimposed onto the ground truth targets. For a majority of predictions, this failed. Here, we only consider the 162 files for finetuned-Boltz and 163 for Boltz-1 where the problem did not appear. For a list of the PDB IDs of these files see Data availability.

[3] the information on Boltz queries can be found in the Boltz repository under `https://github.com/jwohlwend/boltz`

[4] as described at `https://github.com/jwohlwend/boltz/blob/main/docs/evaluation.md`

[5] `https://huggingface.co/boltz-community/boltz-1`

In total, 7 different metrics were calculated. These are:

- rmsd: root mean square deviation of the distances between the superimposed predicted C-alpha atoms and the target C-alpha atoms according to the formula $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2}$
- average distance: the average distance between a predicted water molecule atom and the closest target water molecule atom $\frac{1}{|PW|}\sum_{w_p \in PW} d(w_p, w_t)$ with PW being the set of predicted waters and w_t being the closest target water to w_p
- water rmsd: the root mean square deviation of the distance between a predicted water molecule atom and the closest target water molecule atom $\sqrt{\frac{1}{|PW|}\sum_{w_p \in PW} d(w_p, w_t)^2}$ with PW being the set of predicted waters and w_t being the closest target water to w_p
- share < 0.5: the share of predicted water molecule atoms that are within 0.5 Å distance of a target water molecule atom
- share < 1.0: the share of predicted water molecule atoms that are within 1.0 Å distance of a target water molecule atom
- share < 5.0: the share of predicted water molecule atoms that are within 5.0 Å distance of a target water molecule atom
- share < 10.0: the share of predicted water molecule atoms that are within 10.0 Å distance of a target water molecule atom

The average and standard deviation of the measures for the predictions with 30 water molecules and as many water molecules as in the target file are presented in the following tables:

Predictions with 30 water molecules				
measure	modified-Boltz		Boltz-1	
	average	standard deviation	average	standard deviation
rmsd	3.8979	5.6266	2.8438	5.0403
average distance	8.6969	8.0055	9.4698	9.7671
water rmsd	9.4546	8.2874	10.5778	11.6852
share < 0.5	0.0025	0.0088	0.0018	0.0076
share < 1.0	0.0163	0.0286	0.0147	0.0308
share < 5.0	0.4103	0.2691	0.3736	0.2559
share < 10.0	0.7636	0.3216	0.7395	0.3105

Predictions with as many water molecules as the target				
measure	modified-Boltz		Boltz-1	
	average	standard deviation	average	standard deviation
rmsd	4.0296	6.0025	2.9326	5.1180
average distance	8.8956	9.2449	10.9088	14.4170
water rmsd	9.5783	9.6654	11.9737	15.1077
share < 0.5	0.0010	0.0037	0.0013	0.0051
share < 1.0	0.0116	0.0177	0.0108	0.0202
share < 5.0	0.3892	0.2747	0.3214	0.2096
share < 10.0	0.7708	0.3259	0.6990	0.3120

Chapter 5

Conclusion

5.1 Comparison between Boltz-1 and modified-Boltz

In the tables in 4.4 we can see that modified-Boltz performs noticeable worse than Boltz-1 in rmsd. This means that modified-Boltz is worse in backbone prediction. The other measures, however, are all water-specific and in almost all of these modified-Boltz slightly outperforms the Boltz-1 baseline. However, the performance of modified-Boltz is still not good. This shows us that it is possible to improve water molecule position prediction, however there still is room for improvement.

5.2 Limitations

This work shows a first proof of concept but in that it still has many limitations. The definition of a hydrogen bond was very simple and does not reflect the full nature of a hydrogen bond. For example, bond angles were not considered in this. This also led to some water oxygen atoms having up to 8 hydrogen bonds which is unrealistic.
The failure to train the confidence model was another limitation of this work. With a trained confidence model it would be easier to understand the predictions modified-Boltz made. Also, there was no evaluation during the training process. That means that the model was trained for a certain fixed amount of time rather than as long as it took to reach some performance threshold. With this it was also not possible to check whether the model has been undertrained. Also, because gradient was accumulated over 128 batches, there were only a limited number of gradient steps. This slowed down training.
Another issue were the corrupted predictions mentioned in "Problems during

evaluation inference" in the appendix that have not properly been addressed. Furthermore, the comparability between modified-Boltz and Boltz-1 was limited due to the frequent failings of the superimposition. For this reason, both were evaluated on slightly different data. Also, sometimes the proteins themselves have not been modeled correctly by both models. In these cases, the measures for water have little meaning, yet they still influenced the evaluation metrics.
Another limitation is the model's inability to predict the right amount of water molecules, as this has to be manually set as part of the query.
Finally, a main limitation is the limited performance in the evaluation metrics. Modified-Boltz is not yet able to predict water molecule positions with a good degree of accuracy.

5.3 Outlook

There is both immediate and long-term work that can be done to both improve the performance of modified-Boltz and to strengthen its evaluation.
First of all, I can check other checkpoints that were created during the training. This way it is possible to check whether the model is indeed undertrained by examining how much the performance has improved over the last epochs. Furthermore, I can change the number of water ligands in the queries. This will show whether there is a threshold of the number of water molecules up until which the model performs well. To better evaluate the model, a new evaluation dataset could also be created. This way, I can specifically choose only training data containing single-chain proteins and water molecules for simplicity and comparability. Also, I can avoid the samples where superimposition fails, increasing the comparability of modified-Boltz and Boltz-1.
A deeper case-by-case examination of modified-Boltz's predictions could also be beneficial. This way, specific issues in its predictions can be found. Another possibility is to adapt the data preparation pipeline, especially concerning hydrogen bonds. With this new data, a retraining could be done. For this, the loss function, which the model uses for prediction, could be adapted to also take water-specific measures into account. Also, retraining allows to train the confidence model that has been overlooked before. With the new evaluation, the training model could be validated more thoroughly at each training epoch to better understand the training process.

Data availability

Code that was used for dataset exploration, dataset curation, evaluation, the training configuration file, and pdf of this thesis can be found in the github repository `https://github.com/Maxisman/Boltz_Water_Prediction` or on the USB stick that comes with this thesis.

Acknowledgments

First and foremost, I would like to thank Aleksandr Zlobin who has provided me with this interesting topic and guided me through it. He was always there to answer my questions, track my progress and help me with organizational issues and I am very grateful for his commitment and support.
Also, I want to thank Vsevolod Viliuga and Rajarshi SinhaRoy,who have also helped me a lot with the work on my thesis, especially with the computational part, including the training of the network.
Furthermore, I'd like to thank the Institute for Drug Discovery for providing me with the resources needed to work on my thesis, like a workstation.
Finally, I want to acknowledge that computations for this work were done both using resources of the Leipzig University Computing Center and using the NHR high performance computing Center of TU Dresden. This center is jointly supported by the Federal Ministry of Education and Research and the state governments participating in the NHR (www.nhr-verein.de/unsere-partner).

Appendix

Versions

During the work on this project, a new major Boltz version, Boltz-2, was released on 2025-07-15. The main advancement of Boltz-2 over Boltz-1 is its improved ability to predict binding affinity, which is a measure of how tightly small molecules can attach to proteins. This is of major importance for drug development because it helps to understand how well a therapeutic drug may bind to its intended target and thus, how significant the therapeutic effect will be. Also, Boltz-2 can not only predict static complexes but also dynamic ensembles. Furthermore Boltz-2 reaches a higher level of physical grounding and an improved accuracy across many modalities over Boltz-1. [27]
However, as of the time I worked on data curation, there were not yet the updated scripts for custom data preparation in the official Boltz repository [1]. However, data curation was a major part of this work. Therefore Boltz-1 was still used for this work. To be precise, I took the last commit in the repository before the Boltz-2 commit, commit 85f853d9ee905debc7b59c99b796bc927c22cba0 which was from 2025-05-14. The Boltz software from this commit was used in this project for dataset curation, training and prediction.
For manipulating coordinate files I used ProDy 2.5.0 as it was the most recent version of ProDy when I started my work.

Hardware

The training was done in two parts on two different high-performance computing (HPC) clusters. The first one was Leipzig University Computing Center. There I used partition "clara" with 4 of its Nvidia Tesla V100 GPUs. This hardware includes both CUDA GPU cores and tensor cores especially designed for working with neural networks. The second HPC cluster was from TU Dresden's center for information services and high performance computing (ZIH),

[1] https://github.com/jwohlwend/boltz

where I used the Capella partition that provided Nvidia H100 GPUs of which I used one. These are also especially designed for neural networks and are more powerful than the V100s.

Problems during evaluation inference

During the inference process for model evaluation, an unexpected problem appeared. In some instances, a newline character appeared within a predicted atom line. These newlines appeared either between the values for the _atom_site.auth_comp_id and _atom_site.B_iso_or_equiv values or between the values for the _atom_site.B_iso_or_equiv and _atom_site.pdbx_PDB_model_num attributes. This problem made it impossible for ProDy to parse the .cif file. Subsequently, I removed all newline characters if they were between a line that starts with "ATOM" or "HETATM" and a line that starts with the character 1, as both _atom_site.B_iso_or_equiv and _atom_site.pdbx_PDB_model_num are always 100.000 and 1 respectively for structures predicted by Boltz. However, I was not able to find the source of this problem.

Bibliography

1. Chai, J., Zeng, H., Li, A. & Ngai, E. W. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications* **6,** 100134 (2021).
2. Khurana, D., Koli, A., Khatter, K. & Singh, S. Natural language processing: state of the art, current trends and challenges. *Multimedia tools and applications* **82,** 3713–3744 (2023).
3. Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T. R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., *et al.* Probabilistic weather forecasting with machine learning. *Nature* **637,** 84–90 (2025).
4. Huang, B., Kong, L., Wang, C., Ju, F., Zhang, Q., Zhu, J., Gong, T., Zhang, H., Yu, C., Zheng, W.-M., *et al.* Protein structure prediction: challenges, advances, and the shift of research paradigms. *Genomics, proteomics & bioinformatics* **21,** 913–925 (2023).
5. Pearce, R. & Zhang, Y. Toward the solution of the protein structure prediction problem. *Journal of Biological Chemistry* **297.** ISSN: 0021-9258. `https://doi.org/10.1016/j.jbc.2021.100870` (July 2021).
6. Rohl, C. A., Strauss, C. E., Misura, K. M. & Baker, D. in *Methods in enzymology* 66–93 (Elsevier, 2004).
7. Das, R. & Baker, D. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.* **77,** 363–382 (2008).
8. Fujitsuka, Y., Takada, S., Luthey-Schulten, Z. A. & Wolynes, P. G. Optimizing physical energy functions for protein folding. *Proteins: Structure, Function, and Bioinformatics* **54,** 88–103 (2004).
9. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.* Highly accurate protein structure prediction with AlphaFold. *nature* **596,** 583–589 (2021).

10. Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., *et al.* Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630,** 493–500 (2024).

11. Wohlwend, J. *et al.* Boltz-1 Democratizing Biomolecular Interaction Modeling. *bioRxiv.* eprint: `https://www.biorxiv.org/content/early/2025/05/06/2024.11.19.624167.full.pdf`. `https://www.biorxiv.org/content/early/2025/05/06/2024.11.19.624167` (2025).

12. Mattos, C. Protein–water interactions in a dynamic world. *Trends in Biochemical Sciences* **27,** 203–208. `https://doi.org/10.1016/S0968-0004(02)02067-4` (Apr. 2002).

13. Davies, G. & Henrissat, B. Structures and mechanisms of glycosyl hydrolases. *Structure* **3,** 853–859. ISSN: 0969-2126. `https://doi.org/10.1016/S0969-2126(01)00220-9` (Sept. 1995).

14. Okada, T., Fujiyoshi, Y., Silow, M., Navarro, J., Landau, E. M. & Shichida, Y. Functional role of internal water molecules in rhodopsin revealed by X-ray crystallography. *Proceedings of the National Academy of Sciences* **99,** 5982–5987 (2002).

15. Park, S. & Seok, C. GalaxyWater-CNN: Prediction of Water Positions on the Protein Structure by a 3D-Convolutional Neural Network. *Journal of Chemical Information and Modeling* **62,** 3157–3168. ISSN: 1549-9596. `https://doi.org/10.1021/acs.jcim.2c00306` (July 2022).

16. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. The protein data bank. *Nucleic acids research* **28,** 235–242 (2000).

17. Zardecki, C., Dutta, S., Goodsell, D. S., Lowe, R., Voigt, M. & Burley, S. K. PDB-101: Educational resources supporting molecular explorations through biology and medicine. *Protein Science* **31,** 129–140 (2022).

18. Burley, S. K., Bhatt, R., Bhikadiya, C., Bi, C., Biester, A., Biswas, P., Bittrich, S., Blaumann, S., Brown, R., Chao, H., *et al.* Updated resources for exploring experimentally-determined PDB structures and Computed Structure Models at the RCSB Protein Data Bank. *Nucleic acids research* **53,** D564–D574 (2025).

19. Chari, A. & Stark, H. Prospects and limitations of high-resolution single-particle cryo-electron microscopy. *Annual Review of Biophysics* **52,** 391–411 (2023).

20. Yang, Z., Zeng, X., Zhao, Y. & Chen, R. AlphaFold2 and its applications in the fields of biology and medicine. *Signal Transduction and Targeted Therapy* **8,** 115 (2023).

21. Edgar, R. C. & Batzoglou, S. Multiple sequence alignment. *Current Opinion in Structural Biology* **16.** Nucleic acids/Sequences and topology, 368–373. ISSN: 0959-440X. https://www.sciencedirect.com/science/article/pii/S0959440X06000704 (2006).

22. Ju, F., Zhu, J., Shao, B., Kong, L., Liu, T.-Y., Zheng, W.-M. & Bu, D. CopulaNet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction. *Nature Communications* **12,** 2535. ISSN: 2041-1723. https://doi.org/10.1038/s41467-021-22869-8 (May 2021).

23. Moult, J., Pedersen, J. T., Judson, R. & Fidelis, K. A Large-Scale Experiment to Assess Protein Structure Prediction Methods. *Proteins: Structure, Function, and Bioinformatics* **23.** https://onlinelibrary.wiley.com/doi/10.1002/prot.340230303 (1995).

24. Moult, J. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Current opinion in structural biology* **15,** 285–289 (2005).

25. Yim, J., Stärk, H., Corso, G., Jing, B., Barzilay, R. & Jaakkola, T. S. Diffusion models in protein structure and docking. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **14,** e1711 (2024).

26. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B. & Yang, M.-H. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Comput. Surv.* **56.** ISSN: 0360-0300. https://doi.org/10.1145/3626235 (Nov. 2023).

27. Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., *et al.* Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv,* 2025–06 (2025).

28. Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K. & Ranganath, R. A General Framework for Inference-time Scaling and Steering of Diffusion Models. https://arxiv.org/abs/2501.06848 (2025).

29. Bakan, A., Meireles, L. M. & Bahar, I. ProDy: Protein Dynamics Inferred from Theory and Experiments. *Bioinformatics* **27,** 1575–1577. ISSN: 1367-4803. eprint: https://academic.oup.com/bioinformatics/article-pdf/27/11/1575/48863254/bioinformatics_27_11_1575.pdf. https://doi.org/10.1093/bioinformatics/btr168 (Apr. 2011).

30. Laage, D., Elsaesser, T. & Hynes, J. T. Water Dynamics in the Hydration Shells of Biomolecules. *Chemical reviews* **117,** 10694–10725 (16 2017).

31. Prilusky, J., Hodis, E., Canner, D., Decatur, W., Oberholser, K., Martz, E., Berchanski, A., Harel, M. & Sussman, J. Proteopedia: A status report on the collaborative, 3D web-encyclopedia of proteins and other biomolecules. *Journal of Structural Biology* (Apr. 2011).

32. Mlynek, G., Djinović-Carugo, K. & Carugo, O. B-Factor Rescaling for Protein Crystal Structure Analyses. *Crystals* **14,** 443 (2024).

33. Harding, M. M. The architecture of metal coordination groups in proteins. *Biological Crystallography* **60,** 849–859 (2004).

34. Lipscomb, W. N. & Sträter, N. Recent Advances in Zinc Enzymology. *Chemical Reviews* **96.** PMID: 11848831, 2375–2434. eprint: `https://doi.org/10.1021/cr950042j`. `https://doi.org/10.1021/cr950042j` (1996).