PNAS RESEARCH ARTICLE | PHYSICS OPEN ACCESS

APACE: AlphaFold2 and advanced computing as a service for accelerated discovery in biophysics

Hyun Park[a,b,c], Parth Patel[a,d,e], Roland Haas[e], and E. A. Huerta[a,f,g,1]

Edited by Herbert Levine, Northeastern University, Boston, MA; received August 14, 2023; accepted December 25, 2023

The prediction of protein 3D structure from amino acid sequence is a computational grand challenge in biophysics and plays a key role in robust protein structure prediction algorithms, from drug discovery to genome interpretation. The advent of AI models, such as AlphaFold, is revolutionizing applications that depend on robust protein structure prediction algorithms. To maximize the impact, and ease the usability, of these AI tools we introduce APACE, AlphaFold2 and advanced computing as a service, a computational framework that effectively handles this AI model and its TB-size database to conduct accelerated protein structure prediction analyses in modern supercomputing environments. We deployed APACE in the Delta and Polaris supercomputers and quantified its performance for accurate protein structure predictions using four exemplar proteins: 6AWO, 6OAN, 7MEZ, and 6D6U. Using up to 300 ensembles, distributed across 200 NVIDIA A100 GPUs, we found that APACE is up to two orders of magnitude faster than off-the-self AlphaFold2 implementations, reducing time-to-solution from weeks to minutes. This computational approach may be readily linked with robotics laboratories to automate and accelerate scientific discovery.

AI for science | biophysics | supercomputing | automation

Significance

We introduce APACE, AlphaFold2 and advanced computing as a service, a computational framework that optimizes AlphaFold2 to run at scale in high-performance computing platforms, and which effectively handles this TB-size AI model and database. We showcase the use of APACE in the Delta and Polaris supercomputers to accelerate protein structure prediction for a variety of proteins, and demonstrate that using 200 ensembles distributed over 300 NVIDIA A100 GPUs, APACE reduces time-to-insight from days to minutes. This framework may be readily linked with self-driving laboratories to enable automated discovery at scale.

Innovation at the interface of AI and advanced computing is enabling breakthroughs in science and engineering (1–7). The rise of AI models such as GPT-4 (8), AlphaFold (9), among others, provides new capabilities to accelerate and automate scientific discovery. However, some of these models have not been released to the public, breaking a strong tradition in the AI community. It has been argued that the sheer size of these AI models prohibits their use by a large cross-section of potential users.

To address this shortcoming, we demonstrate how to combine large AI models with high-performance computing platforms to empower a broad cross-section of users to fully exploit the capabilities of AI for scientific discovery. We have selected AlphaFold2 (10) as the science driver for this study, since this AI model is revolutionizing discovery in biophysics, and its use for accurate and rapid protein structure prediction (PSP) demands an optimal use of modern supercomputing environments. Here, we demonstrate how to optimize AlphaFold2 and its database, which exceed 2.6 TB in data storage, to reduce the time needed for accurate PSPs from weeks to minutes.

AlphaFold2's Features. With a deep learning technique–based structure prediction, AlphaFold2 (9) showed unprecedented performance at the 14th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction (11), later improved further with multimer prediction (12). Ever since, efforts to make AlphaFold2 faster (13, 14), to predict protein complex, e.g., antibody (10, 15), and to sample diverse protein conformations (16–18) have come to fruition. In this study, we use AlphaFold2 version 2.3.0., as of August, 2023. The pre-trained neural network parameters used include both monomer and multimer v3.

AlphaFold2 utilizes Central Processing Units (CPUs) to compute key input features: multiple sequence alignment (MSA), and structural templates. MSA represents a collection of protein sequence homologues related to the query protein. MSA captures evolutionary relationships between various proteins such as conserved and variation amino acid residues. MSA is computed using CPU-based sequence alignment algorithms, such as Jackhmmer, which align a query protein sequence with known sequence homologues obtained from databases such as Uniclust. AlphaFold2 can glean key residue interactions from MSA (9). Furthermore, structural templates refer to experimentally known protein homologue structures that share significant sequence similarity with the query protein. These template structures are used to improve the accuracy of AlphaFold2's predictions. CPU-based algorithms, such as HHsearch (for monomer) and Hmmsearch (for multimer), search public protein structure databases such as the Protein Data Bank

Author affiliations: [a]Data Science and Learning Division, Argonne National Laboratory, Lemont, IL 60439; [b]Theoretical and Computational Biophysics Group, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801; [c]Center for Biophysics and Quantitative Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801; [d]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801; [e]National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801; [f]Department of Computer Science, The University of Chicago, Chicago, IL 60637; and [g]Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Author contributions: E.A.H. designed research; H.P., P.P., and R.H. performed research; H.P., P.P., and E.A.H. contributed new reagents/analytic tools; H.P., P.P., and E.A.H. analyzed data; and H.P., P.P., and E.A.H. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: elihu@anl.gov.

Published June 24, 2024.

(PDB). Then, AlphaFold2 extracts spatial information from the template relevant to the query protein (9).

In the Graphics Processing Unit (GPU) phase, AlphaFold2 utilizes the features generated from MSA and templates, passing them through the evoformer network. Evoformer refines representations for both the MSA and pair interactions while iteratively exchanging information between them in a criss-cross fashion to extract amino acid residue relationships. The updated representations then enter a structure module, where predictions for rotations and translations are made to position each residue (9).

The resulting predicted 3D structure undergoes a relaxation process via minimization by Molecular Dynamics (MD) engine to enhance accuracy. Upon generating the final structure, the information cycles back to the beginning of the evoformer blocks in a recycle procedure, further refining the structure predictions. Overall, AlphaFold2 is trained end-to-end, leading to remarkable accuracy and reliability in predicting protein 3D structures (9).

APACE's Improvement over AlphaFold2. We introduce APACE, AlphaFold2 and advanced computing as a service, a computational framework to accelerate AlphaFold2 through CPU & GPU optimizations, and distributed computing in supercomputing environments. Key features of this approach encompass:

Data management. First, APACE facilitates the usage of AlphaFold2's 2.6 TB AI model and database (9) by hosting it at the Delta and Polaris supercomputers (19). AlphaFold2's neural networks can readily access data by leveraging solid state drive (SSD) data storage, and Infinite Memory Engine (IME) data staging.

CPU optimization. Second, inspired by ref. 13, APACE uses Ray library's (20) CPU optimization to parallelize CPU intensive MSA and template computation calculations. As part of CPU optimization, APACE allocates higher CPU cores to MSA/template search tools, rather than default numbers (4 or 8), which showed heuristic speed improvement in our experiments. In addition, in a similar manner as ref. 13, we also implemented a checkpoint to circumvent redundant MSA/template steps if features.pkl file exists, i.e., an intermediate file storing MSA/template search result.

GPU optimization. Third, APACE uses Ray library's (20) GPU optimization to parallelize GPU intensive neural network protein structure prediction steps. An important key difference from ParaFold (13) in terms of GPU speedup is that, ParaFold predicted one conformation with only model_1 (a template-based pretrained model) mostly on peptide sequences (e.g., with an average size less than 100 amino acid residues), rather than protein sequences (i.e., $\sim$ 400 amino acid residues and more). In stark contrast, APACE predicts multiple protein conformations for each protein sequence, and peptide sequence if necessary, with all five pretrained models in parallel, which is computationally demanding.

New functionalities. Fourth, APACE can predict multiple monomer conformations per pretrained neural network model (out of five models), a simple functionality existing only in multimer prediction in the original AlphaFold2 model. APACE includes functionalities such as enabling dropout during structure prediction, changing number of Evoformer (9) recycles, or subsampling MSA options, as provided in ref. 14.

Results and Discussions

We completed three computational experiments to carry out a detailed comparison between APACE and the original AlphaFold2 model. We describe each experiment at a time and then provide the corresponding results.

These results were obtained using the Delta and Polaris supercomputers, housed at the National Center for Supercomputing Applications, and at the Argonne Leadership Computing Facility (ALCF), respectively. Both machines provide highly capable GPU-focused compute environment for GPU and CPU workloads.

Delta offers a mix of standard and reduced precision GPU resources, as well as GPU-dense nodes with both NVIDIA and AMD GPUs. It also provides high-performance node-local SSD scratch file systems, as well as both standard Lustre and relaxed-POSIX parallel file systems spanning the entire resource. On the other hand, the Polaris supercomputer has 560 nodes. Each compute node consists of 1 AMD EPYC Milan processor, four NVIDIA A100 GPUs, unified memory architecture, two fabric endpoints, and two NVMe SSDs. The system interconnect is HPE Slingshot 11, and uses a Dragonfly topology with adaptive routing.

We compared APACE and AlphaFold2 performance using both NVIDIA A100 and A40 GPUs in Delta, and NVIDIA A100 GPUs in Polaris. The computational benchmarks we report below in terms of CPU and GPU runtimes were extracted from the generated timings.json file of both APACE and AlphaFold2.

Experiment 1: Predicting Structures for Four Benchmark Proteins. Four proteins were selected as benchmarks to assess the effectiveness and operational proficiency of APACE. To predict protein structures with APACE, we developed scientific software that enables users to provide suitable headers in sbatch scripts and to load the appropriate environment and module, that are used to successfully submit and complete simulations in the Delta and Polaris supercomputers. These are the Simple Linux Utility for Resource Management (SLURM) parameters we used: --mem=240g, --nodes=10, --exclusive, --ntasks-per-node=1, --cpus-per-task=64, --gpus-per-task=4, --gpus-per-node=4. The neural network and MSA/template-related parameters were the same as AlphaFold2.

Monomers. We used the monomer protein 6AWO (serotonin transporter) as a basic structure to test baseline prediction accuracy and conformational diversity using a total of five models. Thus, we created a Ray cluster consisting of eight NVIDIA A100/A40 GPUs (equivalent to 2 A100/A40 GPU nodes in Delta and Polaris) to facilitate both CPU and GPU parallel execution and relaxation for all five models, i.e., one structure per model, as in the case of AlphaFold2.

Multimers. For multimer proteins, we tested 6OAN, Duffy-binding protein bound with single-chain variable fragment antibody (15); 7MEZ, phosphoinositide 3-kinase (10); and 6D6U, a three distinct chain heteropentamer GABA transporter, which represents a more challenging case for multimer prediction. For each of these proteins, we had eight structure predictions per model, yielding a total of 40 predictions (five ensemble modes $\times$ eight predictions per model). To facilitate concurrent execution and relaxation for the entire array of 40 models, 40 NVIDIA A100 and also 40 A40 GPUs (10 A100/A40 GPU nodes) were harnessed using a Ray cluster.

To initiate a Ray cluster utilizing compute nodes (as described in Methods), we first fetched a list of available compute nodes and their IP addresses. We then launched a head Ray process using one of these nodes, referred to as the "head node." Subsequently, we started Ray worker processes for the remaining compute

Table 1. Performance benchmarks between off-the-shelf AlphaFold2 and our APACE CPU & GPU optimized framework for four exemplar proteins

Protein	# of ensembles	AlphaFold2-A40 CPU/GPU [min]	AlphaFold2-A100 CPU/GPU [min]	APACE-A40 CPU/GPU [min]	APACE-A100 CPU/GPU [min]	AlphaFold2-Polaris CPU/GPU [min]	APACE-A100 Polaris CPU/GPU [min]
1. 6AWO	5	33.0/17.7	33.0/12.8	16.1/4.0	16.1/2.9	189.0/46.8	92.2/9.4
2. 6OAN	40	99.1/268.1	99.4/181.4	56.5/7.9	57.1/5.6	306.2/593.5	175.9/14.8
3. 7MEZ	40	100.7/3756.3	100.7/2339.4	58.0/100.1	58.8/63.0	556.2/3640.9	324.8/91.0
4. 6D6U	40	143.6/1528.7	143.8/786.9	89.3/72.2	89.4/35.1	485.9/1279.1	302.1/32.0

We present results for two types of GPUs available in the Delta supercomputer, NVIDIA A40 and A100 GPUs. We also present results using the Polaris supercomputer housed at the ALCF.

nodes. Each worker is equipped with all four GPUs, and Ray automatically determines the utilization of available GPUs for running and relaxation of models. The workers are then linked to the head node by providing the head node's address.

We utilize srun via message passing interface (MPI) to start the workers on the compute nodes. This is necessary because the sbatch script executes solely on the first compute node. Given the simultaneous launch of all Ray processes using MPI, we incorporate safeguards to prevent race conditions. The race condition safeguards ensure that the head node is started before the worker nodes and the beginning of predictions.

After the underlying Ray cluster was ready, we established a connection to it using ray.init within the run_alphafold.py code and initiated the prediction of the protein structure using APACE. Ray automatically allocates resources and concurrently executes MSA tools on CPUs, model runs, and model relaxation on a distinct GPU. This operation efficiently harnesses the full computational potential of both CPU cores and GPUs available on the compute node.

Experiment 1: Results and Discussion.

CPU acceleration. Through the implementation of parallel optimization techniques, APACE achieved an 1.8X average CPU speedup in Delta, and 1.78X average CPU speedup in Polaris. These results are independent of the number of compute nodes.

GPU acceleration. APACE achieves significant GPU speedups. The following results were obtained using 8 GPUs for 6AWO, and 40 GPUs for 6OAN, 7MEZ, and 6D6U:

1. 6AWO. 4.4× speedup on both A40 and A100 GPUs for Delta; and 4.98× speedup on Polaris.
2. 6OAN. 34× and 32.4× speedup on A40 and A100 GPUs, respectively, for Delta; and 40.1× speedup for Polaris.
3. 7MEZ. 37.5× and 37.1× speedup on A40 and A100 GPUs, respectively, for Delta; and 40X speedup for Polaris.
4. 6D6U. 21.2× and 22× speedup on A40 and A100 GPUs, respectively, for Delta; and 40× speedup for Polaris.

We summarize these results in Table 1. We also note that prediction times are consistently shorter when using NVIDIA A100 GPUs. In brief, APACE provides remarkable speedups for basic and complex structures, retaining the accuracy and robustness of the original AlphaFold2 model. Furthermore, APACE can readily be used for analyses at scale using hundreds of GPUs, as shown below.

Experiment 2: Predicting Protein 7MEZ using 100 and 200 NVIDIA A100 GPUs. To quantify the performance and scalability of APACE in the Delta and Polaris supercomputers, we conducted protein 7MEZ predictions utilizing a significant number of compute nodes. Specifically, we utilized 100 NVIDIA A100 GPUs, which correspond to 25 A100 GPU compute nodes to generate predictions (20 predictions per model). Likewise, we leveraged the computational power of 200 NVIDIA A100 GPUs, equivalent to 50 A100 compute nodes to generate a total of 200 predictions (40 predictions per model). To predict the structures, the sbatch script was modified to allocate the correct number of compute nodes. We also modified the srun and singularity run parameters to successfully complete these calculations.

Experiment 2: Results and Discussion. APACE delivered remarkable speedups. If we compute 100 ensembles (distributed over 100 GPUs) for protein 7MEZ, APACE completed the required calculations within 67.8 min, as opposed to AlphaFold2's 6068.8 min (101.1 h/4.2 d) in Delta. In Polaris, we observe that APACE reduced time-to-solution from 8793.3 min (146.5 h/6.1 d) to 87.9 min.

Similarly, if we now require 200 ensembles for the same protein, APACE in Delta completed all predictions within 64 min, as opposed to the 12023.3 min (200.4 h/8.3 d) that would be needed using the original AlphaFold2 method. In Polaris, APACE reduced time-so-solution from 12741.2 min (212.4 h/8.8 d) to only 84.9 min.

Finally, using 300 ensembles for protein 7MEZ, APACE in Delta completed all predictions within 68.2 min, as opposed to the 18064.3 min (301.1 h/12.5 d) that would be needed using the original AlphaFold2 method. In Polaris, APACE reduced time-so-solution from 15295.6 min (254.9 h/10.6 d) to only 76.9 min. These results are summarized in Table 2.

Experiment 3: Ensemble Diversity of APACE. AlphaFold2's inherent limitations restrict us to generating merely five predictions per monomer, e.g., one prediction per model, thereby

Table 2. Performance benchmarks between off-the-shelf AlphaFold2 and APACE for protein 7MEZ

Nodes/ensembles	AlphaFold2-Delta CPU/GPU [min]	APACE-Delta CPU/GPU [min]	AlphaFold2-Polaris CPU/GPU [min]	APACE-Polaris CPU/GPU [min]
25/100	100.7/6068.8	58.8/67.8	556.2/8793.3	324.8/87.9
50/200	100.7/12023.3	58.8/64.0	556.2/12741.2	324.8/84.9
75/300	100.7/18064.3	58.8/68.2	556.2/15295.6	324.8/76.9

We present results for 25, 50, and 75 nodes in Delta and the Polaris supercomputers. Each node has 4 NVIDIA A100 GPU.

confining the diversity of protein conformation. Moreover, fine-tuning parameters such as dropout remain inaccessible. However, we successfully addressed this constraint by adapting the ColabFold (14) code. For experimental purposes, we generated 100 structures for protein 6AWO using --num_multimer_predictions_per_model=20 while employing the parameter --use_dropout=True. This was accomplished by configuring the sbatch script with the appropriate parameters.

APACE enables users to select the following options (14):

1. Ensemble of structure module with -num_ensemble,
2. Control for recycles with --num_recycles,
3. Subsampling of MSA with --max_seq, --max_extra_seq,
4. Evoformer fusion with --use_fuse,
5. Bfloat16 mixed precision with --use_bfloat16,
6. Bernoulli-masking based diverse conformational sampling with --use_dropout.

Experiment 3: Results and Discussion.

Protein structure prediction and conformational diversity by APACE. We have modified AlphaFold2 code to mirror ColabFold (14)'s versatile protein structure prediction pipeline parameter customization. With these improvements, we have successfully expanded the spectrum of predictions, thereby enhancing the overall reliability of the predicted structures. Although protein structure prediction is of great significance, we would like to expand APACE to predict conformational diversity since proteins are not static but malleable and flexible structures. Sampling a wide range of conformational ensemble is important for drug discovery (16–18). In the case of 6AWO (∼500 amino acid residues), Fig. 1, we used our parameter customization enhancements (with the option --use_dropout=True) and predicted 100 structures of serotonin transporter (SERT). We have found that the structure predicted by APACE is comparable to the ground truth structure. When we visualize most variant transmembrane domain alpha helices (cyan in the *Right* panel), we observe that TM2, TM6, TM10, and TM12 are highlighted. Among these, TM6, TM10, and TM12 are responsible for conformational change or ligand binding from outward-facing to inward-facing structures (22–24). This implies that APACE learned patterns to predict a wide range of conformational landscape of SERT. APACE makes an accurate prediction for SERT, which is an integral membrane protein. Even without the presence of membranes, APACE manages to predict transmembrane domains with high accuracy, hence demonstrating APACE's promise in drug discovery research. In the case of 6OAN, 7MEZ, and 6D6U (∼600, 2,000, and 1,800 amino acid residues, respectively), we have multimer predictions. Both 6OAN and 7MEZ in Figs. 2 and 3 each predict conformational ensemble of heterodimer structures with high accuracy. Especially, the interface binding pose is well predicted and comparable with ground truth structures. Although there may be minor errors in predicted secondary structures not involved in interface binding, correct interface

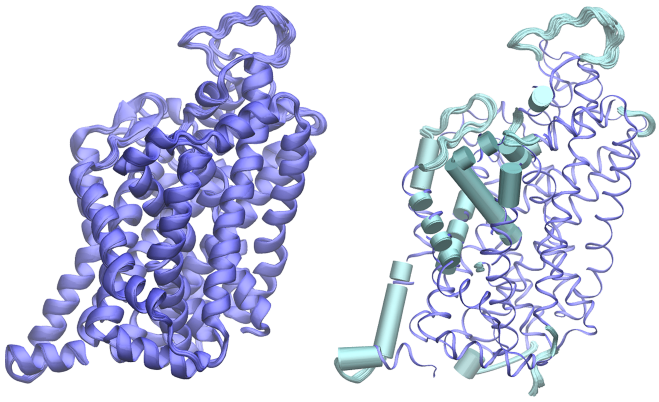

Fig. 1. Protein structure used to test APACE: serotonin transporter (PDB accession: 6AWO; shorthand SERT). The *Left* panel is 100 SERT predicted conformational ensemble overlaid, which has good agreement with ground truth SERT. The *Right* panel is high variant transmembrane domains, shown in cyan, and computed with root mean square fluctuations overlayed. Figures are generated with Visual Molecular Dynamics (21).

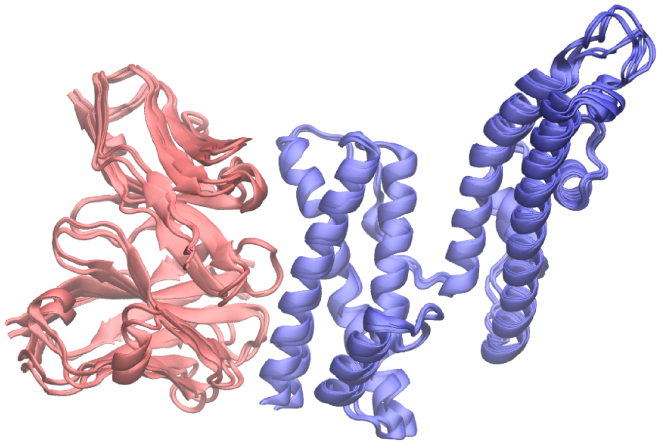

Fig. 2. Protein structure used to test APACE: the antibody–antigen complex Plasmodium vivax Duffy-binding protein (PDB accession: 6OAN). The structure has good agreement with ground truth bound structure conformation. The predicted conformational ensemble of complementary determining region (CDR; loops) of the antibody (red) binding against helical secondary structure epitopes of antibody (blue) are predicted well when compared to ground truth.

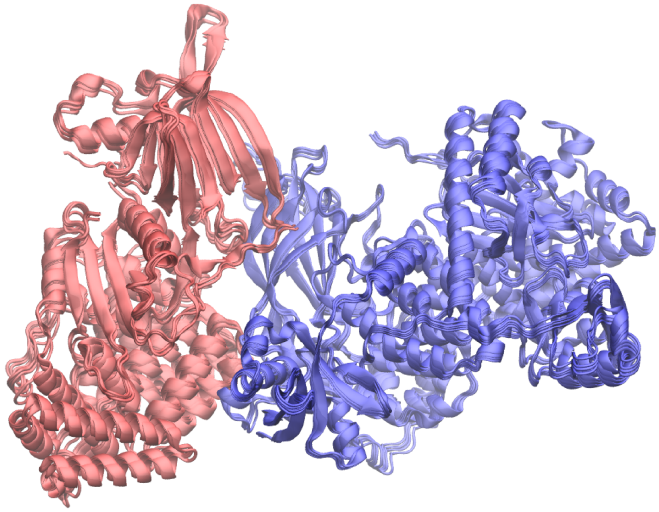

Fig. 3. Protein structure used to test APACE: a phosphoinositide 3-kinase (PI3K) consisting of p110γ and p101 subunits (PDB accession: 7MEZ). The structure has good agreement with ground truth bound structure conformation. Although there are mispredictions of loop secondary structures in p101 (red; *Top Left* helical loop; mispredicted as alpha helix rather than loop) subunit, the interface binding pose between p101 and p110γ (blue) is well predicted, implying conserved binding interface in evolution. Also, rest of the secondary structures and overall heterodimer structure of the predicted conformational ensemble are comparable with ground truth structure.

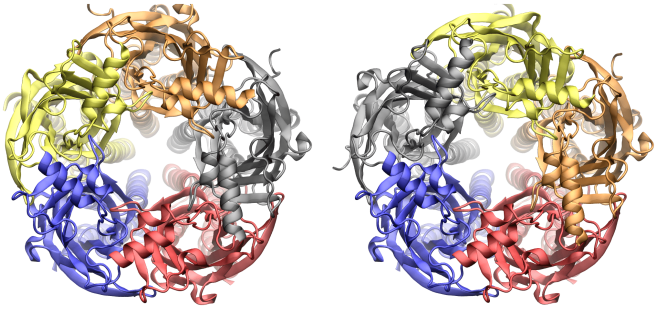

Fig. 4. Protein structure used to test APACE: a pentameric $GABA_A$ receptor (PDB accession: 6D6U). We show one predicted heteropentamer structure of neurotransmitter $GABA_A$ receptor. The *Left* panel shows a comparable structure with ground truth predictions. Blue and gray chains form a homodimer while red and orange chains form the other homodimer. Yellow chain is a monomer differing in sequence from other two homodimers. However, the location of transmembrane helices (toward the paper direction) does not exactly reproduce the ground truth structure. This is understandable since APACE does not use membrane as an input to predict the transmembrane domain. However, the overall structure is comparable with the ground truth. On the other hand, in the *Right* panel, we see an AI-predicted protein whose structure is erroneous, and where blue and gray chains bind to each other. This structure may have high thermodynamic instability and steric hindrance when being crystallized.

binding pose between proteins by nonbonded interactions is of greater importance. Minor misfolding may be addressed with methods such as MD, Monte Carlo, and protein design tools (25–38).

In 6D6U in Fig. 4, we observe comparable structure (*Left* panel) with ground truth and wrong structure (*Right* panel) with wrong homodimer location predictions. Since 6D6U is a membrane pentameric heteromer protein, it lends a challenging case of predicting not only correct structure of each monomer but also alternating chain patterns. The transmembrane helices are therefore mispredicted but overall structure is still comparable with ground truth.

In short, we have demonstrated APACE's capabilities to predict protein structures, mirroring AlphaFold2's robustness and accuracy, and providing remarkable speedups, reducing time-to-solution from days to minutes. APACE may be limited occasionally when it comes to predicting transmembrane proteins and/or multi-chain multimers, features it has inherited from AlphaFold2.

Methods

Given that Delta and Polaris's container support is only available for Apptainer/Singularity (39), we modified the instructions provided in AlphaFold2 GitHub repository, which are intended for Docker containers (40). Below we describe the steps followed to deploy AlphaFold2 on Delta and Polaris (19):

1. We began by cloning the AlphaFold2 directory from DeepMind and navigated to the respective directory. The code is available at https://github.com/hyunp2/alphafold/tree/main.
2. Next, we downloaded the necessary genetic databases for MSA capturing sequence representation, as well as for template similarity capturing structure representation. We also downloaded AlphaFold2 model parameters required for AlphaFold2's functioning. Every database and model parameter is up-to-date (as of August 2023) and the multimer version is v3.
3. Since Singularity is used on Delta and Polaris for container support, we built the Singularity image by initially building the Docker image locally. Afterward, we pushed this Docker image to DockerHub. Utilizing the singularity pull command, we converted the Docker image to Singularity sif format, making it compatible with Delta and Polaris environments.

   Running the Singularity image built using the default Dockerfile resulted in an HHsearch (used for template search against PDB database) runtime error. To address this issue, modifications were made to the Dockerfile. Initially, the Dockerfile involved cloning and compiling the HHsuite package from source locally, which posed portability challenges across different machines. The compilation process with cmake relied on the processor architecture of the user's machine, potentially leading to compatibility issues. For instance, if the user building the Docker image locally had a processor with an Instruction Set Architecture (ISA) that differed from Delta's supported architecture, HHsearch encountered a runtime error with "Illegal Instruction."

   To address this issue and ensure cross-machine compatibility, we made modifications to the DockerFile. Instead of compiling HHsuite from source, we adopted a different approach by installing HHsuite using a statically compiled version that supports the AVX2 ISA. This modification eliminated the dependency on local processor architecture during the build process, mitigating the potential runtime errors and enhancing the portability of the Singularity image.
4. To complete the setup, we created an output directory (default is /tmp/alphafold) and ensured that it had the necessary permissions to allow writing.
5. Due to Delta and Polaris's absence of Docker support, the standard run_docker.py script was not viable. Instead, we devised a custom shell script to replicate the essential functionality of run_docker.py. Employing a singularity run command, we effectively bound the necessary mounts and passed the required flags for execution, mirroring the procedure of run_alphafold.py with Docker.
6. Upon completion of the deployment process, the output directory contained the predicted structures of the target protein, accurately obtained through AlphaFold2's advanced prediction capabilities.

In conclusion, deploying AlphaFold2 on Delta and Polaris required a series of modifications to account for Singularity containerization. Through this approach, we successfully integrated AlphaFold2's powerful protein folding prediction capabilities into these supercomputers' environments.

Limitations of AlphaFold2 Model. The original AlphaFold2 model, while highly accurate in predicting protein structures, does have some limitations in terms of computational efficiency. Some of the key limitations include:

1. Long Inference Time: The time taken for the model to make predictions can be considerable, especially for larger and more complex protein structures. This can hinder its use in time-sensitive applications. Such long time inference including both CPU and GPU computations have been reported and analyzed elsewhere in the literature (13).
2. Computationally Intensive/Limited Real-Time Predictions: The original AlphaFold2 model is computationally demanding, requiring significant computational resources and time for accurate predictions. For example, MSA and template search have to be performed in CPU while GPU is utilized for each structure prediction, both in a sequential manner. This restricts its applicability for real-time predictions or on hardware with limited computational power. The computational demands of the model may hinder real-time prediction of protein structures, making it less suitable for time-sensitive applications.
3. Resource-Intensive: AlphaFold2's inference requires substantial computational resources, including powerful GPUs or tensor processing units. This may limit its accessibility to researchers or institutions without access to high-end hardware. Also, the storage of database amounts to 2.6 TB, which may far exceed normal workstation storage capacity.
4. Memory Requirements: The model's memory footprint can be substantial (with larger protein requiring higher GPU memory), making it challenging to process multiple protein structures concurrently, particularly on machines with limited RAM.
5. Single GPU Utilization: The original AlphaFold2 model is designed to use a single GPU during inference, limiting its capability to work with multiple GPUs. As a result, it predicts and relaxes one protein structure (saved as PDB file format) at a time sequentially.

6. Other potential limitations include but are not limited to little protein conformation diversity. Predicting correct, yet diverse protein conformations, is a significant task for drug discovery, partially addressed in refs. 17 and 16.

Key Optimizations in APACE. To transcend the limitations of AlphaFold2, we implemented optimizations in both CPU and GPU computation, striving for enhanced efficiency and performance.

CPU optimization. AlphaFold2 utilizes Jackhmmer to conduct MSA searches on Uniref90, clustered MGnify and small Bulk File Distribution (BFD) database. On the other hand, it employs HHBlits for MSA search on large BFD and Uniclust30 databases. In addition, AlphaFold2 utilizes HHSearch for template search against PDB70 in the monomer case and Hmmsearch for template search against PDB Seqres in the multimer case. For the multimer only, Jackhmmer based Uniprot database parsing step exists as well.

To process a single query, AlphaFold2 limits itself to 8 CPU cores for Jackhmmer, four CPU cores for HHblits, and eight CPU cores for Hmmsearch. Given the vast database sizes (around 2.6 TB) and the considerable amount of I/O access involved, the MSA search for a single prediction can take several hours, significantly impacting the overall runtime (see *Limitations of AlphaFold2 Model* for bottlenecks for CPU computation).

To expedite the CPU stage, we implemented an approach inspired by ParaFold (13). By orchestrating the three independent sequential MSA searches in parallel, APACE significantly enhances the speed of MSA construction. In contrast to AlphaFold2, where UniRef90, MGnify, and BFD datasets were parsed sequentially using Jackhmmer and/or HHblits (for large BFD), APACE employed the Ray library (20) to simultaneously initiate three processes, enabling MSA searches to run concurrently.

Additionally, we allocated 16 CPU cores to each MSA search tool and template search tool: Jackhmmer, HHblits, and HHSearch/Hmmsearch (monomer/multimer). By running all three MSA tools in parallel, utilizing a total of 48 CPU cores, we achieved a substantial 1.8× speedup in performance.

After increasing the number of CPU cores, we observed remarkable speed enhancements in the MSA computation. However, beyond 20 CPU cores, the speed-up in MSA calculation plateaued. This observation unveiled the bottleneck as being related to input retrieval rather than CPU processing, rendering it input-bound. This inherent nature of being input-bound presents hurdles for straightforward parallelization (i.e., CPU multiprocessing or MPI) methods. Upon the completion of parallel computation for MSA, the template search and multimer Jackhmmer-Uniprot, sequentially ensue.

To further enhance speed in CPU intensive MSA computation, we made two key optimizations. First, we migrated the entire dataset to SSD, minimizing data retrieval time. Additionally, we leveraged the IME to stage the dataset files into the SSD cache within the /ime file system. This pre-staging allowed jobs to swiftly access and utilize the required data. IME is a DataDirect Networks solution designed to facilitate fast data tiering between compute nodes and a file system within a high-performance computing (HPC) environment.

The MSA and structural template search results acquired on CPUs are stored in features.pkl and passed to the neural network for prediction on GPUs. Additionally, we have incorporated a code check in our pipeline to circumvent CPU-burdening MSA computation. That is, if features.pkl already exists (as a result of storing features.pkl by successfully executing CPU computation at least once for a given protein sequence), the pipeline skips the MSA and structure template search and computation steps and proceeds directly to predict the protein structure, in a similar manner as ref. 13. This optimization ensures efficient processing and avoids redundant computations.

GPU optimization. AlphaFold2/APACE employs an ensemble of five neural network models to predict the 3D structure of proteins. This ensemble approach entails using multiple pretrained models with slight variations hyperparameters for protein structure prediction. The three models out of five make predictions based on MSA (i.e., models 3 to 5) while the other two models (i.e., models 1 to 2) also rely on templates. For details of how the five models differ, we refer the refer to refs. 9 and 18

The "--num_multimer_predictions_per_model" flag governs the number of independent predictions made by each individual neural network model within the model ensemble. When running AlphaFold2/APACE, users can specify the value for this flag, thereby controlling the number of predictions generated by each model. The collective predictions from each model in the model ensemble offer diverse final predicted 3D structures, e.g., plasmepsin II, an aspartic protease causing malaria (16), which are crucial to understand free energy landscape of protein conformations and to identify important drug discovery target, cryptic binding pockets. In contrast to AlphaFold2, we included in APACE a capability to predict multiple monomer structures per model.

The original AlphaFold2 model is designed to use a single GPU during inference, which does not take full advantage of deep learning's parallel processing capabilities. During GPU utilization, AlphaFold2 performs sequential structure prediction, which is one of the reasons why AlphaFold2 takes a long time till completion (13) (*Limitations of AlphaFold2 Model*). To expedite the GPU phase, we used the Ray library for GPU parallelization as well for APACE. Therefore, each ensemble model and its corresponding predictions are allocated to distinct GPUs for structure prediction. As a result, APACE can harness multiple GPUs to efficiently run models in parallel, markedly expediting the overall prediction process.

Following prediction by each model, the corresponding structure undergoes a relaxation process in a sequential manner in AlphaFold2. To enhance the efficiency of this step, we once more harnessed the power of the Ray library in APACE. Through this optimization, each structure predicted by ensemble of models is assigned to an individual dedicated GPU, facilitating parallel relaxation processing. This enhancement by APACE has substantially reduced processing time, contributing to the overall acceleration of the relaxation process.

Conclusions

We have introduced APACE, a framework that retains the robustness and accuracy of AlphaFold2, and which leverages supercomputing to reduce time-to-insight from days to minutes. We have accomplished this by a) making an efficient use of the Delta and Polaris supercomputer systems' data storage and data staging; b) optimizing CPU and GPU computing; and c) developing scientific software to enable the prediction of conformational ensemble of protein structures. These tools are released with this manuscript to provide researchers with a computational framework that may be readily linked with robotic laboratories to automate and accelerate scientific discovery.

Data, Materials, and Software Availability. The data and scientific software needed to reproduce this work are available at https://github.com/hyunp2/alphafold/tree/main (41).

ACKNOWLEDGMENTS. This work was supported by Laboratory Directed Research and Development funding from Argonne National Laboratory, provided by the Director, Office of Science, of the United States (U.S.) Department of Energy under Contract No. DE-AC02-06CH11357. E.A.H. was partially supported by NSF award OAC-2209892. This research used resources of the Argonne Leadership Computing Facility, which is a Department of Energy (DOE) Office of Science User Facility supported under Contract DE-AC02-06CH11357. This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. The authors acknowledge support from the National Institute of General Medical Sciences of the National Institutes of Health under awards P41-GM104601, R24-GM145965, and R01-GM123455. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

1. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
2. E. A. Huerta *et al*., Enabling real-time multi-messenger astrophysics discoveries with deep learning. *Nat. Rev. Phys.* **1**, 600–608 (2019).
3. M. Krenn *et al*., On scientific understanding with artificial intelligence. *Nat. Rev. Phys.* **4**, 761–769 (2022).
4. S. Issue, A machine-intelligent world. *Science* **381**, 136–137 (2023).
5. S. Bianchini, M. Müller, P. Pelletier, Artificial intelligence in science: An emerging general method of invention. *Res. Policy* **51**, 104604 (2022).
6. K. Crawford, *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence* (Yale University Press, New Haven, 2021).
7. J. Dean, A golden decade of deep learning: Computing systems & applications. *Daedalus* **151**, 58–74 (2022).
8. OpenAI (2023). GPT-4 technical report.
9. J. Jumper *et al*., Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
10. P. Bryant, G. Pozzati, A. Elofsson, Improved prediction of protein-protein interactions using alphafold2. *Nat. Commun.* **13**, 1265 (2022).
11. J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, M. Topf, Critical assessment of techniques for protein structure prediction, fourteenth round. CASP 14 Abstract Book (2020).
12. R. Evans *et al*., Protein complex prediction with alphafold-multimer. bioRxiv (2021). https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2 (Accessed 30 November 2023).
13. B. Zhong *et al*., "Parafold: paralleling alphafold for large-scale predictions" in *International Conference on High Performance Computing in Asia-Pacific Region Workshops* (Association for Computing Machine, New York, NY, 2022), pp. 1–9.
14. M. Mirdita *et al*., ColabFold: Making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
15. R. Yin, B. Y. Feng, A. Varshney, B. G. Pierce, Benchmarking alphafold for protein complex modeling reveals accuracy determinants. *Protein Sci.* **31**, e4379 (2022).
16. A. Meller, S. Bhakat, S. Solieva, G. R. Bowman, Accelerating cryptic pocket discovery using alphafold. *J. Chem. Theory Comput.* **19**, 4355–4363 (2023).
17. B. Faezov, R. L. Dunbrack Jr., Alphafold2 models of the active form of all 437 catalytically-competent typical human kinase domains. bioRxiv (2023). https://www.biorxiv.org/content/10.1101/2023.07.21.550125v1 (Accessed 30 November 2023).
18. D. Sala, F. Engelberger, H. Mchaourab, J. Meiler, Modeling conformational states of proteins with alphafold. *Curr. Opin. Struct. Biol.* **81**, 102645 (2023).
19. J. Towns *et al*., XSEDE: Accelerating scientific discovery. *Comput. Sci. Eng.* **16**, 62–74 (2014).
20. P. Moritz *et al*., "Ray: A distributed framework for emerging AI applications" in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)* (USENIX Association, USA, 2018), pp. 561–577.
21. W. Humphrey, A. Dalke, K. Schulten, VMD: Visual molecular dynamics. *J. Mol. Graph.* **14**, 33–38 (1996).
22. J. A. Coleman *et al*., Serotonin transporter-ibogaine complexes illuminate mechanisms of inhibition and transport. *Nature* **569**, 141–145 (2019).
23. M. C. Chan, E. Procko, D. Shukla, Structural rearrangement of the serotonin transporter intracellular gate induced by thr276 phosphorylation. *ACS Chem. Neurosci.* **13**, 933–945 (2022).
24. M. C. Chan, B. Selvam, H. J. Young, E. Procko, D. Shukla, The substrate import mechanism of the human serotonin transporter. *Biophys. J* . **121**, 715–730 (2022).
25. G. Brändén, R. Neutze, Advances and challenges in time-resolved macromolecular crystallography. *Science* **373**, eaba0954 (2021).
26. M. E. Mäeots, R. I. Enchev, Structural dynamics: Review of time-resolved cryo-EM. Acta Crystal. *Sect. D Struct. Biol.* **78**, 927–935 (2022).
27. S. J. Amann, D. Keihsler, T. Bodrug, N. G. Brown, D. Haselbach, Frozen in time: Analyzing molecular dynamics with time-resolved cryo-EM. *Structure* **31**, 4–19 (2023).
28. M. Schmidt, Time-resolved macromolecular crystallography at pulsed X-ray sources. *Int. J. Mol. Sci.* **20**, 1401 (2019).
29. F. Martín-García, E. Papaleo, P. Gomez-Puertas, W. Boomsma, K. Lindorff-Larsen, Comparing molecular dynamics force fields in the essential subspace. *PLoS One* **10**, e0121114 (2015).
30. J. K. Leman *et al*., Macromolecular modeling and design in Rosetta: Recent methods and frameworks. *Nat. Methods* **17**, 665–680 (2020).
31. D. Sala, A. Giachetti, A. Rosato, Insights into the dynamics of the human zinc transporter ZnT8 by MD simulations. *J. Chem. Inf. Model.* **61**, 901–912 (2021).
32. D. Sala, A. Giachetti, A. Rosato, An atomistic view of the YiiP structural changes upon Zinc (II) binding. *Biochimica et Biophysica Acta (BBA)-Gen. Subj.* **1863**, 1560–1567 (2019).
33. Y. Matsunaga, Y. Sugita, Use of single-molecule time-series data for refining conformational dynamics in molecular simulations. *Curr. Opin. Struct. Biol.* **61**, 153–159 (2020).
34. L. Cerofolini *et al*., Integrative approaches in structural biology: A more complete picture from the combination of individual techniques. *Biomolecules* **9**, 370 (2019).
35. J. R. Allison, Computational methods for exploring protein conformations. *Biochem. Soc. Trans.* **48**, 1707–1724 (2020).
36. G. Bussi, A. Laio, Using metadynamics to explore complex free-energy landscapes. *Nat. Rev. Phys.* **2**, 200–212 (2020).
37. B. Webb, A. Sali, Comparative protein structure modeling using modeller. *Curr. Protoc. Bioinform.* **54**, 5–6 (2016).
38. K. W. Kaufmann, G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, J. Meiler, Practically useful: What the Rosetta protein modeling suite can do for you. *Biochemistry* **49**, 2987–2998 (2010).
39. G. M. Kurtzer, V. Sochat, M. W. Bauer, Singularity: Scientific containers for mobility of compute. *PLoS One* **12**, e0177459 (2017).
40. D. Merkel, Docker: Lightweight Linux containers for consistent development and deployment. *Linux J.* **2014**, 2 (2014).
41. H. Park, P. Patel, R. Haas, E. A. Huerta, Data and Software from "APACE: AlphaFold2 as a service for accelerated discovery in biophysics". GitHub. https://github.com/hyunp2/alphafold/tree/main. Deposited 20 April 2023.