*Fold

TorchFold: A PyTorch and NPU–Compatible Reimplementation of AlphaFold3

TorchFold Team

Changping Laboratory, Beijing, China
Huawei Technologies Co., Ltd.

Abstract

AlphaFold3 has unified the modeling of proteins, nucleic acids, and ligands at atomic resolution, but its accessibility remains limited by framework dependencies (JAX) and restrictive weight licensing. We present TorchFold, a complete PyTorch reimplementation of AF3 with three key contributions: (1) a native PyTorch implementation enabling seamless integration with the broader deep learning ecosystem; (2) optimized support for Huawei Ascend NPUs alongside NVIDIA GPUs; and (3) commercially friendly model weights under the MIT license. We demonstrate numerical alignment with the original AF3 on protein monomer and antibody-antigen benchmarks. On FoldBench, our models achieve state-of-the-art or highly competitive performance across the majority of tasks. The GPU and NPU inference code is available at `https://github.com/xxx/TorchFold`. Training code and open weights will be released in the coming weeks.

Correspondence: Mingchen Chen (`mingchenchen@cpl.ac.cn`)
Project Page: `https://xxx`

1 Introduction

AlphaFold2 [1] and AlphaFold3 [2] have revolutionized structural biology by achieving experimental-level accuracy in protein structure prediction. However, the original AF3 implementation presents two barriers to broader adoption: (1) its reliance on JAX limits integration with the PyTorch-dominated deep learning ecosystem, and (2) restrictive licensing on model weights prohibits commercial applications.

Several open-source efforts have emerged to address these limitations, including Boltz-1 [3], Chai-1 [4], HelixFold3 [5], Protenix [6], and OpenFold3 [7]. TorchFold builds upon this ecosystem with a complete PyTorch reimplementation, enabling seamless integration with existing workflows and tools. Beyond framework migration, we extend hardware flexibility by providing native support for both NVIDIA GPUs and Huawei Ascend NPUs, while maintaining full compatibility with the original AF3 architecture and weights. We will additionally release commercially friendly model weights to enable unrestricted research and deployment.

2 Methods

2.1 PyTorch Reimplementation

TorchFold is a complete PyTorch reimplementation of AlphaFold 3, migrating all core components from JAX/Haiku to native `nn.Module`. To ensure numerical alignment with the original implementation, we use full `float32` precision throughout, avoiding mixed-precision artifacts that can cause backbone discontinuities.

2.2 NPU Support

Beyond standard CUDA support, TorchFold is natively deployable on Huawei Ascend NPUs via the CANN backend. Several NPU-specific optimizations are applied:

NPU Fusion Attention. The `NPU_Fusion_Attention` operator supports a wide range of input patterns and can eliminate the need for data movement and transposition operations. It delivers significant improvements in both performance and memory reduction, with the gains being positively and non-linearly correlated with sequence length.

Linear Layer Fusion. Merging certain linear-layer computations reduces the pipeline scheduling overhead caused by repeated operator calls, thereby enabling greater parallelism.

Other optimizations. Replacing equivalent performance-superior operators, minimizing random memory-access operations and allocating contiguous memory spaces. Additional enhancements include Granular CPU affinity binding, Two-level task queue, Thread-Caching Malloc and Expandable segments.

2.3 Weights and Licensing

AF3 Compatibility. TorchFold's architecture is strictly aligned with the official AF3 specification, allowing direct loading of original AF3 checkpoints without modification.

Open Weights. We will release a new set of pretrained weights under the MIT license alongside the training code, enabling commercial use. The model is trained end-to-end on large-scale biomolecular structure data, with training objectives aligned with the original AF3 formulation, including diffusion-based structure generation and confidence estimation. To enhance robustness across diverse biomolecular interaction scenarios, we adopt a multi-stage training strategy that begins with large-scale pretraining and concludes with domain-adaptive refinement.

3 Results

3.1 Numerical Alignment with AF3

To validate that TorchFold faithfully reproduces the original AF3 implementation, we compare outputs using identical AF3 weights on two prediction tasks.

3.1.1 Protein Monomer

To evaluate structural prediction accuracy across varying protein sizes, we selected 9 monomeric proteins spanning 37 to 1491 amino acids. As shown in Table 1, both NPU and GPU backends of TorchFold achieve near-identical confidence scores (pTM, pAE, pLDDT) compared to the original AF3, with low RMSD values relative to experimental structures confirming faithful reproduction of AF3's prediction quality.

Table 1 Numerical alignment between TorchFold and AF3 across different sequence lengths.

Metric	Method	Sequence Length								
		37	**107**	**301**	**436**	**583**	**740**	**1024**	**1303**	**1491**
pTM	AF3	0.69	0.82	0.93	0.91	0.98	0.96	0.96	0.98	0.95
	TorchFold (GPU)	0.67	0.82	0.93	0.90	0.98	0.96	0.96	0.98	OOM
	TorchFold (NPU)	0.68	0.83	0.93	0.90	0.98	0.96	0.96	0.98	0.95
pAE	AF3	3.15	4.55	4.02	4.20	1.86	3.12	4.52	2.49	5.69
	TorchFold (GPU)	2.92	4.65	3.92	4.44	1.85	3.10	4.14	2.51	OOM
	TorchFold (NPU)	2.71	4.50	3.75	4.28	1.84	3.12	4.06	2.48	5.59
pLDDT	AF3	91.60	85.82	91.63	93.14	97.59	95.53	94.60	96.88	90.63
	TorchFold (GPU)	92.94	85.67	91.53	92.56	97.69	95.58	94.92	96.89	OOM
	TorchFold (NPU)	93.40	85.98	91.84	93.00	97.61	95.53	95.05	96.84	90.93
$RMSD_{C\alpha}$	AF3	0.73	0.89	0.31	1.02	0.26	0.25	0.25	0.16	0.62
	TorchFold (GPU)	0.67	0.90	0.31	0.77	0.26	0.27	0.29	0.15	N/A
	TorchFold (NPU)	0.71	0.83	0.30	0.89	0.30	0.25	0.33	0.15	0.61
$RMSD_{all}$	AF3	1.24	1.08	0.45	1.04	0.30	0.29	0.32	0.19	0.68
	TorchFold (GPU)	1.15	1.16	0.49	0.80	0.29	0.33	0.34	0.19	N/A
	TorchFold (NPU)	1.26	1.09	0.49	0.92	0.32	0.29	0.37	0.19	0.67

3.1.2 Antibody–Antigen Complex

We constructed the PDB55 dataset from SAbDab [8] by filtering for high-resolution structures (<4 Å) with valid CDR-antigen contacts. Non-redundancy was ensured by clustering antigen sequences at 95% and 70% identity levels. We identified 48 clusters (98 complexes) with release dates exclusively after May 1, 2025. The final PDB55 set consists of 55 structures after excluding complexes longer than 512 residues.

Model performance was evaluated through extensive stochastic sampling, where 109 random seeds were used per test case, with 5 samples generated per seed (totaling 545 decoys per case). DockQ [9] scores were computed for all generated decoys relative to the ground truth structures. Based on DockQ values, predictions were categorized into three quality levels: Acceptable ($0.23 \leq$ DockQ < 0.49), Medium ($0.49 \leq$ DockQ < 0.80), and High (DockQ ≥ 0.80). Two primary metrics were reported: Top-1 DockQ, representing the quality of the structure with the highest ranking score, and Coverage DockQ, representing the highest DockQ achieved among all inferences for a given case.

As shown in Table 2, TorchFold achieves comparable performance to AF3 across all DockQ thresholds, confirming numerical alignment on antibody-antigen docking tasks.

Table 2 Antibody-antigen docking performance on PDB55. Values indicate number of successful predictions out of 55 cases.

Method	Top-1 DockQ			Coverage DockQ		
	Acceptable	Medium	High	Acceptable	Medium	High
AF3	27/55	25/55	14/55	47/55	33/55	20/55
TorchFold (GPU)	29/55	26/55	14/55	45/55	33/55	20/55
TorchFold (NPU)	27/55	26/55	16/55	45/55	32/55	20/55

3.2 Inference Performance

Table 3 compares inference performance across GPU (NVIDIA A100 80GB), NPU (Atlas 800T A2), and the JAX-based AF3 baseline. The NPU backend achieves competitive latency across all sequence lengths and successfully handles sequences up to 2002 residues, while the GPU version encounters out-of-memory

errors beyond 1024 residues. Ongoing work focuses on parallelizing diffusion sampling and optimizing the featurization pipeline.

Table 3 Inference latency (seconds) across sequence lengths. "OOM": out-of-memory.

Method	37	107	301	436	583	740	1024	2002
AF3 (JAX, A100)	40.62	40.78	54.45	76.59	76.80	77.09	105.08	321.44
TorchFold (GPU, A100)	39.89	66.51	77.51	110.27	164.69	236.54	467.83	OOM
TorchFold (NPU, Atlas)	42.68	43.72	50.40	60.45	79.91	109.46	163.76	828.86
GPU / AF3 Ratio	1.02	0.61	0.70	0.69	0.47	0.33	0.22	–
NPU / AF3 Ratio	0.95	0.93	1.08	1.27	0.97	0.70	0.64	0.39

3.3 Benchmark Comparison

We evaluate TorchFold on FoldBench [10], a standardized benchmark covering diverse biomolecular structure prediction tasks. We compare against state-of-the-art methods including AlphaFold3 [2], Boltz-1 [3], Chai-1 [4], HelixFold3 [5], Protenix [6], and OpenFold3 [7]. Performance metrics for baseline methods are cited directly from FoldBench.

As shown in Table 4, TorchFold with AF3 weights achieves performance on par with the original AlphaFold3 on both protein-protein and antibody-antigen interface prediction tasks. TorchFold with our independently trained weights (open weights) further improves antibody-antigen docking performance, while consistently outperforming other open-source alternatives including Boltz-1, Chai-1, HelixFold3, Protenix, and OpenFold3.

Table 4 FoldBench results on interface prediction tasks. Success rates are reported as DockQ ≥ 0.23.

Model	Protein–Protein	Antibody–Antigen
AlphaFold3	72.93%	47.90%
TorchFold + AF3 weights	72.69%	45.45%
TorchFold + open weights	72.69%	46.67%
Boltz-1	68.25%	33.54%
Chai-1	68.53%	23.64%
HelixFold3	66.27%	28.40%
Protenix	68.18%	34.13%
OpenFold3	69.96%	28.83%

4 Conclusion

TorchFold democratizes access to state-of-the-art structure prediction. By providing a PyTorch-native, NPU-supported, and commercially unrestricted alternative to AlphaFold 3, we empower the community to accelerate discoveries in computational protein design and drug development.

5 Data Availability

Inference Code. The inference code is available at `https://github.com/xxx/TorchFold`, with full documentation and worked examples.

Training Code and Model Weights. The training pipeline (including data processing scripts and loss implementations) along with our open-licensed model weights will be released together in a future update.

6 Acknowledgments

7 Author Contributions

Yu Liu[1,*], Di Wang[1,*], Zhengyi Li[1,*], Shuxian Gao[2,*], Hongzhun Wang[1], Zhouhanyu Shen[1], Zhiqi Ma[1], Yucheng Zhang[2], Mingchen Chen[1,†]

[1]Changping Laboratory, Beijing, China

[2]Huawei Technologies Co., Ltd.

[*]Equal contribution

[†]Corresponding authors

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. 596(7873):583–589.

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. 630(8016):493–500.

[3] Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 Democratizing Biomolecular Interaction Modeling.

[4] Chai Discovery, Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life.

[5] PaddleHelix Team. Technical report of HelixFold3 for biomolecular structure prediction.

[6] ByteDance AML AI4Science Team, Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, Shenghao Wu, Kuangqi Zhou, Yanping Yang, Zhenyu Liu, Lan Wang, Bo Shi, Shaochen Shi, and Wenzhi Xiao. Protenix - Advancing Structure Prediction Through a Comprehensive AlphaFold3 Reproduction.

[7] The OpenFold3 Team. OpenFold3-preview.

[8] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M. Deane. SAbDab: The structural antibody database. 42(D1):D1140–D1146.

[9] Claudio Mirabello and Björn Wallner. DockQ v2: Improved automatic quality measure for protein multimers, nucleic acids, and small molecules. 40(10):btae586.

[10] Sheng Xu, Qiantai Feng, Lifeng Qiao, Hao Wu, Tao Shen, Yu Cheng, Shuangjia Zheng, and Siqi Sun. Benchmarking all-atom biomolecular structure prediction with FoldBench. 17(1):442.