SimpleFold: Folding Proteins is Simpler than You Think

Yuyang Wang Apple **Jiarui Lu** Apple **Navdeep Jaitly** Apple **Josh Susskind** Apple **Miguel Angel Bautista** Apple

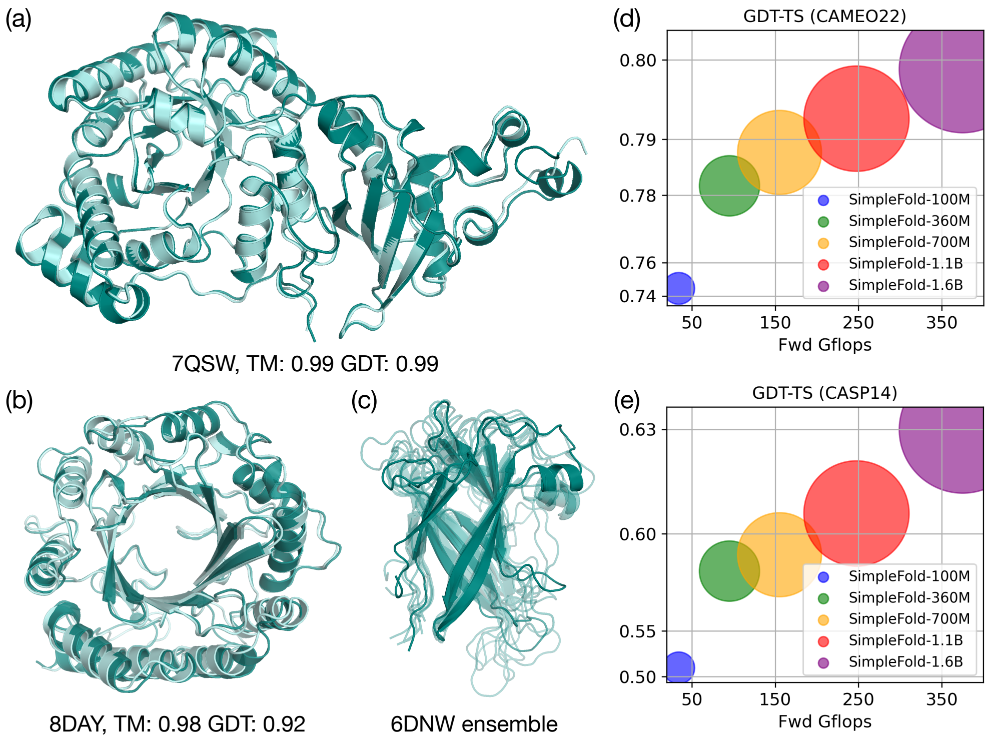
Abstract

Protein folding models have achieved groundbreaking results typically via a combination of integrating domain knowledge into the architectural blocks and training pipelines. Nonetheless, given the success of generative models across different but related problems, it is natural to question whether these architectural designs are a necessary condition to build performant models. In this paper, we introduce *SimpleFold, the first flow-matching based protein folding model that solely uses general purpose transformer blocks*. Protein folding models typically employ computationally expensive modules involving triangular updates, explicit pair representations or multiple training objectives curated for this specific domain. Instead, SimpleFold employs standard transformer blocks with adaptive layers and is trained via a generative flow-matching objective with an additional structural term. We scale SimpleFold to 3B parameters and train it on approximately 9M distilled protein structures together with experimental PDB data. On standard folding benchmarks, SimpleFold-3B achieves competitive performance compared to state-of-the-art baselines, in addition SimpleFold demonstrates strong performance in ensemble prediction which is typically difficult for models trained via deterministic reconstruction objectives. Due to its general-purpose architecture, SimpleFold shows efficiency in deployment and inference on consumer-level hardware. SimpleFold challenges the reliance on complex domain-specific architectures designs in protein folding, opening up an alternative design space for future progress.

1 Introduction

Established protein folding models like AlphaFold2 [23] and RoseTTAFold [8] have achieved groundbreaking accuracy by relying on carefully engineered architectures that integrate computationally heavy domain-specific designs for protein folding tasks such as multiple sequence alignments (MSAs) of AA sequences, pair representations, and triangle updates [23, 8]. These design choices (MSA, pair representations, triangular updates, etc.) are an attempt to hard-code our current understanding of the underlying structure generation process into these models, instead of opting to let models to learn this directly from data, which could be beneficial for a variety of reasons. For example, [27] showed that for orphan proteins (those with few or no close homologs) approaches based on protein language models (PLM) tend to outperform approaches that rely on MSA like AlphaFold2. In this paper, we propose a strong departure from domain-specific designs towards a much more general architectural design which has been demonstrated to be effective in generative modeling problems and can ultimately leverage data and compute as effectively as possible.

In this work, we propose *SimpleFold*, a flow-matching based folding model that directly maps a protein sequence to its full 3D atomic structure without relying on MSA, pairwise interaction maps, triangular updates or any other equivariant geometric modules. Our architecture is inspired by recent transformer-based text-to-image and text-to-3D flow matching models [33, 32], with a strong emphasis on departing from current architecture designs using a general-purpose transformer



7QSW, TM: 0.99 GDT: 0.99

8DAY, TM: 0.98 GDT: 0.92

6DNW ensemble

Figure 1: Example predictions of SimpleFold on targets (a) chain A of 7QSW and (b) chain A of 8DAY, with ground truth shown in light aqua and prediction in deep teal. (c) Generated ensembles of target chain B of 6NDW with SimpleFold finetuned on MD ensemble data. (d) Performance of SimpleFold on CASP14 with increasing model sizes from 100M to 3B. (e) Inference time of different sizes of SimpleFold on consumer level hardware, i.e., M2 Max 64GB Macbook Pro.

backbone trained end-to-end with a flow-matching training objective. Crucially, we demonstrate that strong folding performance (see Fig. 1 can be achieved without explicit pairwise representations, triangle updates, or MSA, which significantly reduces architectural complexity and challenges preconceived notions around the necessity of these designs [27]. *SimpleFold* represents a strong departure from previous of protein folding models, and we summarize our contributions as follows:

- We revisit protein folding as a conditional generative task and introduce SimpleFold, a flow-based transformer folding model that eliminates MSA, pairwise representations, and triangle modules.
- We scale SimpleFold to 3B parameters and train it on approximately 9M distilled structures together with PDB experimental data.
- Our most powerful SimpleFold-3B model shows strong results in folding compared to baselines with hard-coded heuristic designs and also achieves competitive performance on ensemble generation.
- We train a family of models ranging from an efficient 100M model to a large 3B model for the best performance (Fig. 1(d)). SimpleFold-100M recovers ∼90% performance of our best model on major folding benchmarks while being very efficient in inference even on consumer-level devices.

2 SimpleFold

2.1 Folding with Flow-Matching

SimpleFold implements a linear interpolant path [4] (also referred to as a rectified flow [28, 16]) between samples from the empirical data distribution $\mathbf{x} \sim p_{\mathcal{D}}$ and noise samples $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, such that $\mathbf{x}_t = t\mathbf{x} + (1-t)\boldsymbol{\epsilon}$, where the target velocity is defined as $\mathbf{v}_t = \mathbf{x} - \boldsymbol{\epsilon}$. In flow matching, we train a network $\mathbf{v}_\theta$ to match the target across time and data via ℓ_2 regression objective $\mathbb{E}[||\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}_t||^2]$.

In particular, given a protein with N_a heavy atoms, we build a linear interpolant between noise $\boldsymbol{\epsilon}$ and all-atom positions $\mathbf{x}$, where $\boldsymbol{\epsilon}, \mathbf{x} \in \mathbb{R}^{N_a \times 3}$, conditioned on the amino acid sequence $\mathbf{s} \in \mathbb{R}^{N_r}$, where N_r is number of residues or amino acids in the protein. Unlike earlier work that modeled only the $C_\alpha$backbone with flow-matching models [25, 26, 17], we generate full-atom conformations including both backbones and side chains. The network $\mathbf{v}_\theta$ takes the amino acid sequence as a conditioning input $\mathbf{v}_\theta(\mathbf{x}_t, \mathbf{s}, t)$ to model the target velocity field. In particular, the flow-matching objective is defined as follows: $\ell_{\text{FM}} = \mathbb{E}_{\mathbf{x},\mathbf{s},\boldsymbol{\epsilon},t}\left[\frac{1}{N_a}\left\|\mathbf{v}_\theta(\mathbf{x}_t, \mathbf{s}, t) - (\mathbf{x} - \boldsymbol{\epsilon})\right\|^2\right]$.

We also include an additional local distance difference test (LDDT) loss similar to [1]. This loss measures the atomic pairwise distances error between the generated structure $\hat{\mathbf{x}}(\mathbf{x}_t)$ at timestep t and ground truth structures $\mathbf{x}$. During training, $\hat{\mathbf{x}}(\mathbf{x}_t)$ is estimated through one step Euler, i.e., $\hat{\mathbf{x}}(\mathbf{x}_t) = \mathbf{x}_t + (1-t)\,\mathbf{v}_\theta(\mathbf{x}_t, \mathbf{s}, t)$. The LDDT loss is formulated as follows: $\ell_{\text{LDDT}} = \mathbb{E}_{\mathbf{x},\mathbf{s},\boldsymbol{\epsilon},t}\left[\frac{\sum_{i\neq j}\mathbb{1}(\delta_{ij}<\mathcal{C})\sigma(\|\delta_{ij}-\hat{\delta}^t_{ij}\|)}{\sum\mathbb{1}(\delta_{ij}<\mathcal{C})}\right]$, where $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ and $\hat{\delta}^t_{ij} = \|\hat{\mathbf{x}}(\mathbf{x}_t)_i - \hat{\mathbf{x}}(\mathbf{x}_t)_j\|$ denote the distances between atom i, j in ground truth and predicted structures, respectively. The term $\sigma(\cdot)$ is a nonlinear function on pair distance errors and $\mathcal{C}$ is a cutoff distance which controls neighboring atoms to be included in the loss. The model is trained with a weighted combination of flow-matching and LDDT terms: $\ell = \ell_{\text{FM}} + \alpha(t)\ell_{\text{LDDT}}$, where $\alpha(t)$ is a weighting term related to timestep t in flow process and is also dependent to different training phases.

To improve training efficiency and force generating structures with fine details [16, 17], the timestep t is sampled from the distribution: $p(t) = 0.98\texttt{LN}(0.8, 1.7) + 0.02\mathcal{U}(0, 1)$, where $\texttt{LN}$ is logistic-normal distribution [6] and $\mathcal{U}$ is a uniform distribution. We shift the sample weight towards timesteps that are closer to clean data (i.e., $t = 1$), similar to findings in [17]. This improves quality of generated samples especially in modeling refined structures of side chain atoms.

2.2 Architecture

Architectural components like triangle updates and explicit modeling of interactions between single representations and pair representations have been adopted as standard in protein folding models since AlphaFold2 [23] was introduced. It remains an open question whether these architectural design decisions are a necessary condition to build performant models. In a strong departure from previous approaches, SimpleFold uses an architecture solely based on general-purpose transformer modules. In Fig. 2 we show an architecture diagram of SimpleFold, which contains three major modules: light-weighted atom encoder and decoder which are symmetric (i.e., same number of blocks and hidden size) and a heavy residue trunk. All modules are implemented with standard transformer blocks with adaptive layers conditioned on the timestep t (see bottom left of Fig. 2).

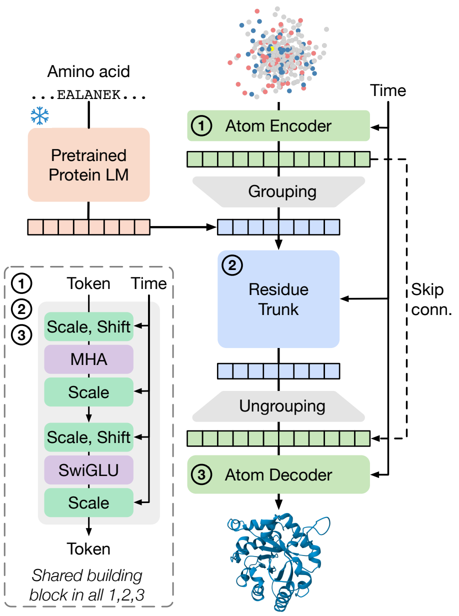


Figure 2: Overview of SimpleFold's architecture built on general-purpose standard Transformer block with adaptive layers. Atom encoder, residue trunk, and atom decoder all share the same general-purposed building block. Our model circumvents the need for pair representations or triangular updates.

Similar to text-to-image and text-to-3D generative models, we use a frozen pretrained protein language model (PLM) to embed the AA sequence into an informative latent representation. We leverage ESM2-3B [27] in all our models to encode the AA sequence $\mathbf{s}$ into per-residue conditioning embeddings $\mathbf{e} \in \mathbb{R}^{N_r \times d_e}$. Sequence embeddings are then concatenated with the residue tokens along the channel dimension and fed into the residue trunk. The residue trunk contains most of the parameters of the model and is where most of the compute is spent on. The grouping operation takes the output of the atom encoder and conducts average pooling to atom tokens within the same residue to obtain residue tokens $\mathbf{r} \in \mathbb{R}^{N_r \times d_a}$. Whereas the ungrouping operation projects residue tokens to corresponding atom tokens.

2.3 Sampling

To fold a protein with a given amino acid sequence $\mathbf{s}$ in inference, we initialize atomic coordinates as Gaussian noise $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and integrate the learned vector field from $t = 0$ to $t = 1$. We perform stochastic generation using a Langevin-style SDE formulation of the flow process [3, 37, 32]:

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_\theta(\mathbf{x}_t, \mathbf{s}, t)\,\mathrm{d}t + \frac{1}{2}w(t)\mathbf{s}_\theta(\mathbf{x}_t, t, c)\,\mathrm{d}t + \sqrt{\tau \cdot w(t)}\,\mathrm{d}\bar{\mathbf{W}}_t, \tag{1}$$

where $w(t) > 0$ is a time-dependent diffusion coefficient, $\bar{\mathbf{W}}_t$ is a reverse-time Wiener process, τ controls the scale of stochasticity, and $\mathbf{s}_\theta(\mathbf{x}_t, \mathbf{s}, t) = (t\mathbf{v}_\theta(\mathbf{x}_t, \mathbf{s}, t) - \mathbf{x}_t)/(1 - t)$. We find $w(t) = \frac{2(1-t)}{t+\eta}$, which defines stochasticity scheduler following SNR of flow process and η is a small constant for numerical stability, gives the best sampling quality. We stick to this setting in all our experiments unless mentioned otherwise. Similar to previous flow-matching based protein generative models [17], we find that τ balances the generation of accurate refined structures and modeling the ensemble of conformations.

2.4 Training Data

We train SimpleFold with a data mix of 3 different sources. First, we include around 160K structures from Protein Data Bank (PDB) [9, 45, 5] with a cutoff of May 2020 following ESMFold [27]. Additionally, we use the SwissProt set from AFDB. Within SwissProt distilled structures, we select samples with average pLDDT greater than 85 and standard deviation of pLDDT smaller than 15, which yields approximately 270K distilled samples. Moreover, we use representative protein structures for each cluster in AFESM [46]. We filter these structures with pLDDT larger than 0.8 resulting in more than 1.9M distilled strictures. To train our biggest model SimpleFold-3B, we explore an extended version AFESM (which we call AFESM-E) by also including structures beyond the cluster representatives. In particular, for each cluster, we randomly pick a maximum of 10 proteins structures with average pLDDT larger than 80, which resulting in a total of 8.6M distilled structures. Since larger models with larger capacity benefit from larger training sets, we train our largest SimpleFold-3B on the distilled AFESM-E data together with PDB and SwissProt.

3 Experiments

We evaluate SimpleFold on two widely adopted protein structure prediction benchmarks: CAMEO22 and CASP14, which are rigorous tests for generalization, robustness, and atomic-level accuracy in folding models. CAMEO22 [18] follows the setting in [22] which contains 183 targets structures. In addition, CASP14 [34] is a more challenging benchmark containing selective targets for a biennial blind prediction challenge. We set $\tau = 0.01$ for SimpleFold in inference which empirically shows best general performance in folding.

Despite its simplicity, SimpleFold achieves competitive performance compared with these baselines. In both benchmarks, SimpleFold shows consistently better performance than ESMFlow which is also a flow-matching model built with ESM embeddings. On CAMEO22, SimpleFold demonstrates comparable results to the best folding models (e.g., ESMFold, RoseTTAFold2, and AlphaFold2). In particular, SimpleFold achieves over 95% performance of RoseTTAFold2/AlphaFold2 on most metrics without applying expensive and heuristic triangle attention and MSA. On the more challenging CASP14 benchmark, SimpleFold achieves even better performance than ESMFold. In particular, SimpleFold-3B obtains a TM-score of 0.720 / 0.792 and GDT-TS of 0.639 / 0.703 in comparison to 0.701 / 0.792 and 0.622 / 0.711 of ESMFold. SimpleFold also shows competitive or even better performance to baselines that applies MSA like RoseTTAFold and AlphaFlow. It is also notable that all models except AlphaFold2 show a significant performance drop on CASP14 compared to CAMEO22, even AlphaFlow which is a finetuned flow-matching model using a pre-trained AlphaFold2 model as initialization. We attribute this to the fact that AlphaFold2 leverages templates from MSA and uses a regression training objective. We note that the performance drop of SimpleFold on CASP14 w.r.t. CAMEO22 is much smaller compared to many baselines model like ESMFold. Given that neither ESMFold or SimpleFold rely on MSA, this demonstrates that SimpleFold is very robust in predicting valid structures on challenging tasks.

Moreover, scaling up the model sizes of SimpleFold models results in better performance across the board, which indicates the benefit of designing a general purpose approach that benefits from scale.

Table 1: Performance of protein folding on the CAMEO22 and CASP14 benchmarks. For each metric, we report the average / median over all samples. Here, orange, green, blue denote baselines trained with regression objectives, generative objectives, and our SimpleFold, respectively.

Type	**Model**	**TM-score** ↑	**GDT-TS** ↑	**LDDT** ↑	**LDDT-C_α** ↑	**RMSD** ↓
			CAMEO22			
MSA-based	RoseTTAFold [8]	0.780 / 0.860	0.715 / 0.775	0.575 / 0.605	0.798 / 0.827	5.721 / 2.864
	AlphaFlow [21]	0.840 / 0.927	0.808 / 0.853	0.741 / 0.798	0.855 / 0.893	3.846 / 2.122
	AlphaFold2 [23]	0.863 / 0.942	0.844 / 0.903	0.816 / 0.856	0.893 / 0.923	3.578 / 1.857
	RoseTTAFold2 [7]	0.864 / 0.947	0.845 / 0.904	0.727 / 0.767	0.893 / 0.926	3.571 / 1.707
PLM-based	ESM3 [19]	0.746 / 0.840	0.694 / 0.758	–	–	–
	ESMDiff [30]	0.754 / 0.847	0.701 / 0.760	–	–	–
	EigenFold [22]	0.750 / 0.840	0.710 / 0.790	–	–	–
	OmegaFold [44]	0.805 / 0.899	0.767 / 0.844	0.746 / 0.815	0.829 / 0.892	5.294 / 2.622
	ESMFlow [21]	0.818 / 0.893	0.774 / 0.832	0.696 / 0.745	0.827 / 0.867	4.528 / 2.693
	ESMFold [27]	0.853 / 0.933	0.826 / 0.875	0.792 / 0.834	0.871 / 0.906	3.973 / 2.019
Ours	SimpleFold-100M	0.803 / 0.878	0.746 / 0.787	0.721 / 0.752	0.822 / 0.852	4.897 / 2.855
	SimpleFold-360M	0.826 / 0.905	0.782 / 0.841	0.773 / 0.803	0.844 / 0.878	4.775 / 2.681
	SimpleFold-700M	0.829 / 0.915	0.788 / 0.845	0.775 / 0.809	0.850 / 0.886	4.557 / 2.423
	SimpleFold-1.1B	0.833 / 0.924	0.793 / 0.851	0.776 / 0.807	0.850 / 0.883	4.350 / 2.334
	SimpleFold-1.6B	0.835 / 0.916	0.799 / 0.864	0.782 / 0.816	0.853 / 0.889	4.397 / 2.187
	SimpleFold-3B	0.837 / 0.916	0.802 / 0.867	0.773 / 0.802	0.852 / 0.884	4.225 / 2.175
			CASP14			
MSA-based	RoseTTAFold [8]	0.654 / 0.678	0.562 / 0.572	0.464 / 0.456	0.705 / 0.723	9.676 / 6.420
	AlphaFlow [21]	0.740 / 0.812	0.661 / 0.711	0.632 / 0.662	0.767 / 0.799	7.091 / 3.949
	RoseTTAFold2 [7]	0.802 / 0.881	0.740 / 0.824	0.638 / 0.669	0.824 / 0.869	6.744 / 3.292
	AlphaFold2 [23]	0.845 / 0.907	0.783 / 0.855	0.778 / 0.817	0.856 / 0.897	5.027 / 3.015
PLM-based	ESMDiff [30]	0.521 / 0.499	0.447 / 0.430	–	–	–
	ESM3 [19]	0.534 / 0.567	0.459 / 0.488	–	–	–
	EigenFold [22]	0.590 / 0.637	0.539 / 0.575	–	–	–
	ESMFlow [21]	0.627 / 0.679	0.539 / 0.544	0.525 / 0.539	0.669 / 0.730	10.503 / 6.974
	OmegaFold [44]	0.693 / 0.773	0.625 / 0.723	0.627 / 0.726	0.715 / 0.824	9.845 / 4.042
	ESMFold [27]	0.701 / 0.792	0.622 / 0.711	0.637 / 0.705	0.725 / 0.802	8.679 / 4.016
Ours	SimpleFold-100M	0.611 / 0.628	0.513 / 0.544	0.537 / 0.549	0.659 / 0.685	11.157 / 8.976
	SimpleFold-360M	0.674 / 0.758	0.585 / 0.654	0.617 / 0.657	0.703 / 0.762	9.382 / 4.828
	SimpleFold-700M	0.680 / 0.767	0.591 / 0.668	0.630 / 0.674	0.714 / 0.763	9.289 / 4.431
	SimpleFold-1.1B	0.697 / 0.796	0.607 / 0.668	0.640 / 0.676	0.723 / 0.758	9.249 / 4.462
	SimpleFold-1.6B	0.712 / 0.801	0.630 / 0.709	0.660 / 0.699	0.741 / 0.798	8.424 / 4.722
	SimpleFold-3B	0.720 / 0.792	0.639 / 0.703	0.666 / 0.709	0.747 / 0.829	7.732 / 3.923

It is notable that scaling up model sizes improves performance substantially more in CASP14, i.e the more challenging benchmark, than in CAMEO22. This is a clear empirical evidence that models with larger capacity are more capable of solving complex folding tasks.

4 Conclusions

We have introduced SimpleFold, a flow-matching based generative model for protein folding that represent a strong departure from the architectural designs in previous approaches. SimpleFold is solely built with general-purpose transformer blocks with adaptive layers, dispensing away with heuristic designs like expensive pair representations and triangular updates introduced by AlphaFold2. We believe SimpleFold represents a disruptive approach for protein folding that relies on scaling up general purpose architecture blocks to learn the symmetries of the underlying data generation process directly from training data.

References

[1] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Zemgulyte, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Zidek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630:493–500, 2024.

[2] G. Ahdritz, N. Bouatta, C. Floristean, S. Kadyan, Q. Xia, W. Gerecke, T. J. O'Donnell, D. Berenberg, I. Fisk, N. Zanichelli, et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature methods*, 21(8):1514–1524, 2024.

[3] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[4] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[5] D. R. Armstrong, J. M. Berrisford, M. J. Conroy, A. Gutmanas, S. Anyango, P. Choudhary, A. R. Clark, J. M. Dana, M. Deshpande, R. Dunlop, et al. Pdbe: improved findability of macromolecular structure data in the pdb. *Nucleic acids research*, 48(D1):D335–D343, 2020.

[6] J. Atchison and S. M. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

[7] M. Baek, I. Anishchenko, I. R. Humphreys, Q. Cong, D. Baker, and F. DiMaio. Efficient and accurate prediction of protein structure using rosettafold2. *BioRxiv*, pages 2023–05, 2023.

[8] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[9] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[10] J. Boitreaud, J. Dent, M. McPartlon, J. Meier, V. Reis, A. Rogozhnikov, and K. Wu. Chai-1: Decoding the molecular interactions of life. *BioRxiv*, 2024.

[11] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[12] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[13] D. Chakravarty and L. L. Porter. Alphafold2 fails to predict protein fold switching. *Protein Science*, 31(6):e4353, 2022.

[14] S. Cheng, X. Zhao, G. Lu, J. Fang, Z. Yu, T. Zheng, R. Wu, X. Zhang, J. Peng, and Y. You. Fastfold: Reducing alphafold training time from 11 days to 67 hours. *arXiv preprint arXiv:2203.00854*, 2022.

[15] D. Del Alamo, D. Sala, H. S. Mchaourab, and J. Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *Elife*, 11:e75751, 2022.

[16] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[17] T. Geffner, K. Didi, Z. Zhang, D. Reidenbach, Z. Cao, J. Yim, M. Geiger, C. Dallago, E. Kucukbenli, A. Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025.

[18] J. Haas, S. Roth, A. Arnold, T. Kiefer, L. Schmidt, L. Bordoli, and T. Schwede. Continuous automated model evaluation (cameo) complementing the critical assessment of structure prediction in casp12. *Proteins: Structure, Function, and Bioinformatics*, 86(S1):387–398, 2018.

[19] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, page eads0018, 2025.

[20] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

[21] B. Jing, B. Berger, and T. Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

[22] B. Jing, E. Erives, P. Pao-Huang, G. Corso, B. Berger, and T. Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

[23] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[24] Z. Li, X. Liu, W. Chen, F. Shen, H. Bi, G. Ke, and L. Zhang. Uni-fold: an open-source platform for developing protein folding models beyond alphafold. *bioRxiv*, pages 2022–08, 2022.

[25] Y. Lin and M. AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

[26] Y. Lin, M. Lee, Z. Zhang, and M. AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

[27] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[28] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[29] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[30] J. Lu, X. Chen, S. Z. Lu, C. Shi, H. Guo, Y. Bengio, and J. Tang. Structure language models for protein conformation generation. *arXiv preprint arXiv:2410.18403*, 2024.

[31] J. Lu, B. Zhong, Z. Zhang, and J. Tang. Str2str: A score-based framework for zero-shot protein conformation sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

[33] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[34] J. Pereira, A. J. Simpkin, M. D. Hartmann, D. J. Rigden, R. M. Keegan, and A. N. Lupas. High-accuracy protein structure prediction in casp14. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1687–1699, 2021.

[35] Z. Qu, R. Chen, D. Xue, X. Zhou, X. Zeng, and Q. Gu. P(all-atom) is unlocking new path for protein design. *bioRxiv*, 2024.

[36] T. Saldaño, N. Escobedo, J. Marchetti, D. J. Zea, J. Mac Donagh, A. J. Velez Rueda, E. Gonik, A. García Melani, J. Novomisky Nechcoff, M. N. Salas, et al. Impact of protein conformational diversity on alphafold predictions. *Bioinformatics*, 38(10):2742–2748, 2022.

[37] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[38] B. A. A. Team, X. Chen, Y. Zhang, C. Lu, W. Ma, J. Guan, C. Gong, J. Yang, H. Zhang, K. Zhang, et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pages 2025–01, 2025.

[39] Y. Vander Meersche, G. Cretin, A. Gheeraert, J.-C. Gelly, and T. Galochkina. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic acids research*, 52(D1):D384–D392, 2024.

[40] Y. Wang, A. A. Elhag, N. Jaitly, J. M. Susskind, and M. A. Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. *arXiv preprint arXiv:2311.17932*, 2023.

[41] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. Vázquez Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[42] J. Wohlwend, G. Corso, S. Passaro, M. Reveiz, K. Leidal, W. Swiderski, T. Portnoi, I. Chinn, J. Silterra, T. Jaakkola, et al. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, pages 2024–11, 2024.

[43] J. Wohlwend, M. Reveiz, M. McPartlon, A. Feldmann, W. Jin, and R. Barzilay. Minifold: Simple, fast, and accurate protein structure prediction. *Transactions on Machine Learning Research*.

[44] R. Wu, F. Ding, R. Wang, R. Shen, X. Zhang, S. Luo, C. Su, Z. Wu, Q. Xie, B. Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pages 2022–07, 2022.

[45] wwPDB consortium. Protein data bank: the single global archive for 3d macromolecular structure data. *Nucleic Acids Research*, 47(D1):D520–D528, 10 2018.

[46] J. Yeo, Y. Han, N. Bordin, A. M. Lau, S. M. Kandathil, H. Kim, E. L. Karin, M. Mirdita, D. T. Jones, C. Orengo, et al. Metagenomic-scale analysis of the predicted protein structure universe. *bioRxiv*, pages 2025–04, 2025.

A Related Work

Protein Folding Since the development of AlphaFold2 [13] and RoseTTAFold [8] which achieved groundbreaking performance in protein folding with learning-based methods, many works have continued to investigate this problem [2, 7, 24]. AlphaFold2 introduced domain specific network modules like triangle attention and design decisions like explicitly modeling interactions between single and pair representations. It also relied on MSA to extract evolutionary information of protein sequences in the hopes to nudge the model towards biological experts understanding of the underlying data generation process.

OmegaFold [44] and ESMFold [27] replaced MSA with learned embeddings from pretrained protein language model, which are efficient in inference and especially beneficial for orphan proteins. Some works also aimed at accelerating the models through efficient implementations of AlphaFold2 modules, like FastFold [14] and MiniFold [43]. These folding models are built on regression objectives of local frame instead of direct modeling of all-atom positions. Therefore structural predictions of these models lack diversity for ensemble generation.

Flow-Matching for Proteins Generative models, especially diffusion and flow-matching based methods, have been introduced to protein folding given its superior performance in generating high-quality plausible samples. AlphaFlow/ESMFlow [21] proposed to tune AlphaFold2/ESMFold with flow-matching objectives and demonstrated advantages in ensemble generation. However, [21] were not build from the ground up as generative models and instead rely on powerful pretrained AlphaFold2 and ESMFold models which were trained with a deterministic regression objective. AlphaFold3 [1] and its architectural reproductions (e.g., Boltz-1 [42], Protenix [38], Chai-1 [10]) also used diffusion to build generative models for protein complexes of biomolecular interactions. In addition, several works have investigated diffusion or flow-matching models for de novo protein structure generation, like RFDiffusion [41], Genie-2 [26], P(all-atom) [35]. Though these works have employed diffusion or flow-matching generative models for proteins, they still heavily rely on heuristic architectural designs from AlphaFold series like expensive triangle attention and explicit modeling of pair representations. Some are also built on crafted equivariant diffusion process [22]. Proteina [17] attempts to build a simplified architecture but still explicitly applies pair representation, and it only models C_α generation. Previously, MCF [40] investigated conformation generation of small molecular systems with general-purpose transformer backbone. In a strong departure from previous protein folding models, SimpleFold aims at tackling the folding problem with a general purpose transformer backbone and learning symmetries in the underlying data generation process directly from training data.

B Model Configurations

Table 2 lists the configurations of different SimpleFold models from the smallest 94M to largest 2.86B. In implementation, we apply the same architecture for the atom encoder and atom decoder. Though AlphaFold2 is similar to our smallest SimpleFold-100M in terms of number of parameters (both are around 95M), its forward Gflops are much higher than our largest SimpleFold-3B ($\sim$ 30Tflops vs. $\sim$ 1.4Tflops). This is because AlphaFold2 relies on expensive triangle update as well as explicit modeling pair representations from MSA. SimpleFold, on the other hand, is built on general-purposed transformer blocks which are much more computationally efficient.

C Experimental settings

We train a family of SimpleFold models at different sizes (i.e., 100M, 360M, 700M, 1.1B, 1.6B, and 3B) to investigate the scaling ability of proposed framework in folding. When scaling up model sizes, we increase the depth and hidden size of atom encoder and decoder as well as residue trunk altogether (see detailed configurations in Tab. 2). During training we copy one protein B_c times per GPU with different flow timestep t sampled and accumulate gradients from B_p different proteins on different GPUs, following AlphaFold2 [13, 1]. Therefore, the effective batch size is $B_c \times B_p$. We empirically find that this strategy leads to a more stable gradient and better performance than naively building a batch with randomly selected proteins.

Table 2: Configurations of different variants of SimpleFold with comparison to AlphaFold2 and ESMFold in number of parameters and forward Gflops.

			Atom Enc. / Dec.			**Residue Trunk**		
Model	# Params	Gflops	Dim.	# Heads	# Blocks	Dim.	# Heads	# Blocks
AlphaFold2	95M	30935.0	-	-	-	-	-	-
ESMFold	710M	3399.7	-	-	-	-	-	-
SimpleFold-100M	94M	66.5	256	4	1	768	12	8
SimpleFold-360M	360M	189.9	256	4	2	1024	16	18
SimpleFold-700M	687M	310.4	256	4	2	1152	16	28
SimpleFold-1.1B	1.11B	496.0	384	6	2	1280	20	36
SimpleFold-1.6B	1.58B	750.0	512	8	3	1536	24	36
SimpleFold-3B	2.86B	1382.4	640	10	4	2048	32	36

Table 3: Evaluation on MD ensembles. Results of baseline models are taken from [21, 30], to which the evaluation pipeline for our SimpleFold (SF) and SimpleFold-MD (SF-MD) adheres.

	No Tuning			*Tuned*			
	AF2	MSA-sub.	SimpleFold	ESMDiff	ESMFlow-MD	AlphaFlow-MD	SimpleFold-MD
Pairwise RMSD r ↑	0.10	0.22	**0.44**	0.18	0.19	**0.48**	0.45
Global RMSF r ↑	0.21	0.29	**0.45**	0.49	0.31	**0.60**	0.48
Per target RMSF r ↑	0.52	0.51	**0.60**	0.68	0.76	**0.85**	0.67
RMWD ↓	**3.58**	4.28	4.22	7.48	3.60	**2.61**	4.17
RMWD trans contri ↓	**2.86**	3.33	3.74	5.18	3.13	**2.28**	3.40
RMWD var contri ↓	2.27	2.24	**1.74**	3.37	1.74	**1.30**	1.88
MD PCA W2 ↓	1.99	2.23	**1.62**	2.29	1.51	1.52	**1.34**
Joint PCA W2 ↓	2.86	3.57	**2.59**	6.32	3.19	**2.18**	2.85
% PC sim > 0.5 ↑	23	21	**37**	23	26	**44**	38
Weak contacts J ↑	0.27	0.37	0.36	0.52	0.55	**0.62**	0.56
Transient contacts J ↑	**0.28**	0.27	**0.27**	0.26	0.34	**0.41**	0.34
Exposed residue J ↑	0.32	0.37	**0.39**	-	0.49	0.50	**0.60**
Exposed MI matrix ρ ↑	0.02	0.10	**0.14**	-	0.20	0.25	**0.32**

Pre-training. The overall training of SimpleFold consistent of two training stages pre-training and finetuning, which only differ on the data used to train the model. During the pre-training stage of SimpleFold we use a large dataset containing as much available data as possible. Finetuning, on the other hand, is performed on high-quality data to increase the fidelity of generated structures. In pre-training, SimpleFold is trained on approximately 2M (8.7M for the 3B model) data structures including all three data sources, namely PDB, SwissProt from AFDB, and AFESM. We set the maximal amino acid sequence length to 256, where we keep shorter sequence without padding while crop longer sequences to 256 residues. We set $\alpha(t) = 1$ which uses LDDT supervision through the whole flow process. All models are trained with effective batch size 512 except for 1.6B and 3B models which are trained with batch size 1024 and 3072, respectively. We use the AdamW optimizer [29] with learning rate 0.0001 and linear warmup for the first 5000 steps.

Finetuning. In finetuning, SimpleFold is trained on PDB and SwissProt subsets only which contain higher quality data. We set a maximal sequence length to 512 which allows access to larger protein structures in this training phase. We accordingly half B_c in each batch to fit in GPU memory. We set $\alpha(t) = 1 + 8\texttt{ReLU}(t - 0.5)$ in which gradually increases weight of LDDT loss to maximum value of 5 when approaching clean data ($t = 1$). We keep AdamW as an optimizer with the same learning rate 0.0001 in finetuning. In both pre-training and finetuning, we apply an exponential moving average (EMA) of all model weights with a decay of 0.999 following a common practice in flow-matching generative models.

D Ensemble Generation

D.1 Molecular dynamic ensemble

SimpleFold trivially models the distribution of protein structures, due its generative training objective. Namely, SimpleFold does not only generate one deterministic structure for an input AA sequence but is also capable of generating the ensemble of different conformations. To demonstrate this ability of SimpleFold, we benchmark the performance on the ATLAS dataset [39], which assess generation of molecular dynamic (MD) ensemble structures. ATLAS contains contains all-atom MD simulations

Table 4: Two-state conformation results. For the last two metrics, both mean and median are reported over the targets. Results are taken from the ESMDiff paper [30], to which the evaluation pipeline for the rest models adhere.

Type	Model	Res. flex. (global) ↑	Res. flex. (per-target) ↑	TM-ens ↑	Res. flex. (global) ↑	Res. flex. (per-target) ↑	TM-ens ↑
			Apo/holo			*Fold-switch*	
Seq-based	FoldFlow2 [20]	0.027	0.057 / 0.055	0.216 / 0.208	0.051	0.009 / 0.005	0.199 / 0.191
	MultiFlow [11]	0.113	0.211 / 0.194	0.360 / 0.342	0.092	0.068 / 0.061	0.269 / 0.250
	Str2Str [31]	0.174	0.326 / 0.307	0.731 / 0.728	0.161	0.246 / 0.233	0.615 / 0.644
	Eigenfold [22]	0.126	0.407 / 0.401	0.830 / 0.870	0.225	0.279 / 0.255	0.614 / 0.653
	ESMDiff [30]	0.420	0.489 / 0.515	0.838 / 0.877	**0.402**	0.341 / 0.288	0.626 / 0.685
	ESMFlow [21]	0.416	0.496 / 0.522	0.856 / 0.893	0.269	0.345 / 0.329	0.700 / 0.755
MSA-based	MSA-Subs. [23]	0.398	0.404 / 0.371	0.856 / 0.894	0.350	0.320 / 0.303	0.714 / 0.765
	AlphaFlow [21]	0.455	0.527 / 0.527	0.864 / 0.893	0.385	**0.384 / 0.376**	**0.730 / 0.788**
Ours	SimpleFold-100M	0.492	0.500 / 0.532	0.852 / 0.887	0.391	0.291 / 0.241	0.656 / 0.677
	SimpleFold-360M	0.537	0.520 / 0.528	0.864 / 0.898	0.359	0.310 / 0.314	0.689 / 0.746
	SimpleFold-700M	0.552	0.524 / 0.538	0.870 / 0.899	0.307	0.328 / 0.310	0.693 / 0.713
	SimpleFold-1.1B	0.557	0.526 / 0.537	0.870 / 0.900	0.337	0.346 / 0.344	0.698 / 0.755
	SimpleFold-1.6B	0.501	0.522 / 0.508	0.877 / 0.912	0.240	0.339 / 0.318	0.721 / 0.770
	SimpleFold-3B	**0.639**	**0.550 / 0.552**	**0.893 / 0.916**	0.292	0.288 / 0.263	**0.734 / 0.766**

of 1390 proteins. We follow AlphaFlow [21] for training, validation, and test split of ATLAS and evaluate generated 250 conformations for each protein in test set. Tab. 3 compares SimpleFold with baseline models on ATLAS. Reported metrics comprehensively measure the quality of generated ensembles from predicting flexibility (e.g., RMSD r and RMSF r), distributional accuracy (e.g., RMWD), and ensemble observables (e.g., exposed residue and exposed MI matrix).

Firstly, we directly evaluate our largest SimpleFold-3B without additional tuning on MD simulation data in ATLAS. We set $\tau = 0.6$ in inference to add more stochasticity than folding tasks. We compare our approach to baseline models, AlphaFold2 [13] and MSA subsampling [15]. MSA subsampling introduces more stochasticity to AlphaFold2 by subsampling the aligned AA sequences from MSA search. Note the ESMFold is trained via a deterministic regression objective, thus cannot be applied to ensemble generation without additional tuning. Compared to baselines, SimpleFold achieves superior performance on generating ensembles that match the distribution from MD simulations.

We also report the results of SimpleFold-MD, a finetuned model on the training data split of ATLAS, comparing to baselines that are also additionally tuned (i.e., ESMDiff [30], ESMFlow-MD [21], and AlphaFlow-MD [21]). In particular, a fully trained SimpleFold is tuned for additional 20K iterations, where we keep $\alpha(t) = 1$. As shown in Tab. 3, SimpleFold consistently achieves better performance than ESMFlow-MD where both rely on the ESM embedding without MSA. SimpleFold also shows better performance than AlphaFlow-MD on metrics related to ensemble observables (e.g., exposed residue and MI matrix), which are a key feature in the identification of cryptic pockets in drug discovery.

D.2 Multi-state structure prediction

We also evaluate the capacity of SimpleFold to generate structures for proteins showing more than one natural conformation. We adopt the benchmarking set of apo-holo conformational change [36] (*Apo/holo*) and fold-switchers [13] (*Fold-switch*) following EigenFold [22]. The target in each dataset is represented by (1) an amino acid sequence and (2) two distinct ground truth structures. The model is required to produce a diverse yet accurate set of samples "covering" both conformational states and reflecting correct local flexibility.

We compare SimpleFold with a collection of existing approaches including both (1) sequence-based approaches: FoldFlow2 [20], MultiFlow [12], Str2Str [31], EigenFold [22], ESMDiff [30] and ESMFlow [21]; (2) MSA-based methods, including MSA subsampling [15] and AlphaFlow [21]. For each dataset, we report the global and per-target residue flexibility (res. flex.), as well as the ensemble TM score (TM-ens) [22]. The evaluation protocol follows previous works [22, 30], where five samples are generated to compute the metrics with respect to the two ground truth conformations for each target. In inference, we empirically set $\tau = 0.8$ for SimpleFold which generates structures that align with both native conformations and correctly model residue flexibility.

As shown in Tab. 4, SimpleFold obtains state-of-the-art performance on Apo/holo, where SimpleFold outperforms strong MSA-based approaches like AlphaFlow significantly. On Fold-switch, SimpleFold shows comparable or even better performance than ESMFlow which is also applies flow-matching objective and is built on ESM embeddings. The results validate the capability of our SimpleFold in predicting the structures of high quality (i.e., ensemble TM-score) as well as correctly modeling the flexibility in structures (i.e., residue flexibility). Also, the overall performance of SimpleFold increases with the model size growing, which further showcase potential of our proposed framework in generating protein ensembles. Experiments on both MD ensemble and multi-state structure benchmarks demonstrate the capability of SimpleFold in modeling the ensemble of protein structures, which can be beneficial for applications that requires flexibility modeling of protein structures (e.g., molecular docking).

E Effects of Scaling in Protein Folding

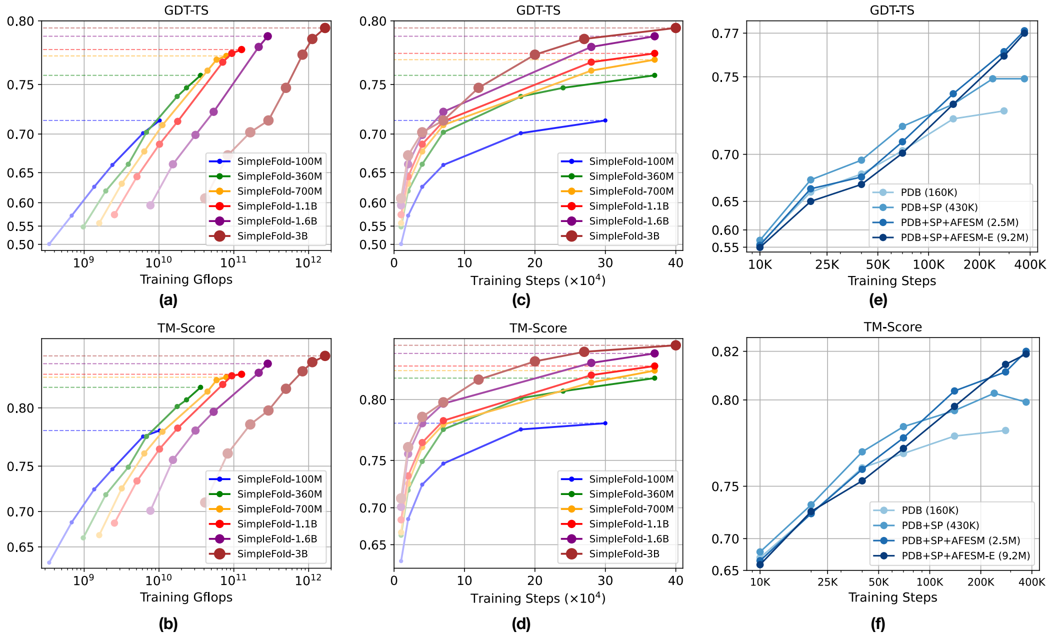


Figure 3: Scaling behavior of SimpleFold. Training Gflops vs. folding performance on GDT-TS and (b) TM-score. Training steps vs. folding performance on (c) GDT-TS and (d) TM-score. How data scale affects the performance (e) GDT-TS and (f) TM-score. All models are benchmarked on CAMEO22.

SimpleFold benefits from increasing model sizes as proven by recent success of generative models in other domains, like vision and language generation. We note that the effects of scaling both training data and model sizes have note yet been rigorously investigated in protein folding. In this section, we empirically show the scaling behavior of SimpleFold from both model and data perspectives, highlighting important considerations for building powerful biological generative models.

To assess the benefit of scaling up the model size in SimpleFold, we train models with different sizes from the smallest with 100M parameters to the largest with 3B parameters. All models are trained with full pre-training data containing PDB, SwissProt from AFDB, and filtered AFESM. Fig. 3(a)-(d) illustrate how model sizes affect the performance of folding (also see Fig. 1(d)). Larger models trained with a larger training budget (i.e., training Gflops and training iterations), are preferred to achieve better performance. We believe these results highlight the positive scaling behavior of SimpleFold and highlight an direction of progress to obtain more powerful generative models in biology.

We also show the benefits of scaling up training data in SimpleFold. We train SimpleFold-700M with different sources of training data: (1) PDB only (160K structures), (2) a combination of PDB and SwissProt (SP, 270K structures) from AFDB, (3) filtered representative proteins from AFESM (1.9M structures) in addition to PDB and SwissProt, and (4) the extended AFESM set (AFESM-E) which contains additional proteins besides the representative protein in each cluster (a total of 8.6M structures). As shown in Fig. 3(e) and (f), SimpleFold as we increase the total number of unique structures in the data mix, the final performance of SimpleFold tends to improve after 400k training iterations. These experimental results support our core contribution to build a simplified and scalable folding model that benefits from the growing total of protein data available either experimentally or distilled from different models.