FrameBench: A Principled, Symmetry-Aware Evaluation of SE(3) Protein Generators

Wenran LI[*]
University of Paris City and University of Reunion
France
li.wenran@univ-reunion.fr

Xavier Cadet[*]
Dartmouth College, Hanover
USA[†]
xavier.cadet.fjf@gmail.com

Cedric Damour
University of Reunion
France
cedric.damour@univ-reunion.fr

Yu LI
Beijing University of Technology
China
yuli@bjut.edu.cn

Alexandre G. de Brevern
University of Paris City and University of Reunion
France
alexandre.debrevern@univ-paris-diderot.fr

Alain Miranville
University Le Havre Normandie
France
alain.miranville@univ-lehavre.fr

Frederic Cadet[‡]
University of Paris City and University of Reunion
PEACCEL, AI for Biologics, Paris
France
frederic.cadet.run@gmail.com

Abstract

We propose an evaluation framework for the $SE(3)$ group symmetry diffusion model and flow matching models: FrameDiff, FrameFlow, and FoldFlow. We show how they expose trade-offs among diversity, domain stability, and compute-normalized efficiency. We instantiate this principle in FrameBench, which treats diversity, stability, and efficiency as quantities of the generated protein. Concretely, FrameBench reports (i) diversity via CATH entropy and coverage; (ii) stability via SWORD2-based domain partitions; and (iii) fairness via compute-normalized efficiency. Evaluating FrameDiff, FrameFlow, and FoldFlow under matched length-/compute budgets reveals clear trade-offs, e.g., coverage versus stability, that simple wall-clock timings miss.

1 Introduction

The structure of a protein is not only a marvel of biological architecture but also the source of its functional power. The long-standing challenge of generating protein structures has been transformed by advances in AI, which can use a generative model to quickly produce accurate and functional proteins Saharkhiz et al. [2024]. Generative models such as diffusion models Ho et al. [2020] and

[*] *Equal contribution*
[†] *Work done whilst at Imperial College London, UK*
[‡] Corresponding Author

EurIPS 2025 Workshop on Principles of Generative Modeling (PriGM).

flow matching models Lipman et al. [2022] have already shown a strong ability to generate novel and diverse protein structures. Especially these kinds of models are good at keeping $SE(3)$ group symmetry, which is perfectly adapted to the property of protein backbone.

The diffusion model Ho et al. [2020] consists of two parts, which can be represented as a Stochastic Differential Equation (SDE) Song et al. [2020]. (1) The first is the forward process, which controls the noise scaling; (2) the second is the denoising process, whose purpose is to learn the previously mentioned noise magnitude in order to generate a pretrained model. For all diffusion processes, there exists a corresponding deterministic process whose trajectories share the same marginal probability densities as the SDE.

Diffusion models have shown remarkable success in generating realistic protein structures from Gaussian noise. Representative examples such as AlphaFold3 Abramson et al. [2024], Chroma Ingraham et al. [2023], and RFDiffusion Watson et al. [2023] illustrate the strong potential of diffusion-based approaches in capturing complex structural distributions. FrameDiff Yim et al. [2023b], the model evaluated in our study, follows this line of work and is likewise designed to generate accurate distributions over protein conformations.

Flow matching models Lipman et al. [2024], based on Ordinary differential equations, represent the data as vector fields. Given a source distribution p and a target distribution q, flow matching is a scalable approach for training a flow model, defined by a learnable velocity u_t^θ, and solving the flow matching problem: Find u_t^θ, with $p_0 = p$ and $p_1 = q$.

In this paper, we focus on two flow-matching–based generative models for protein backbone generation. (1) FrameFlow follows the same architectural design as FrameDiff but replaces diffusion with deterministic flow matching. It is used for unconditional backbone generation and also extends naturally to constrained tasks such as motif scaffolding. (2) FoldFlow is a recently proposed family of $SE(3)$ flow-matching models that further improves stability and sampling quality by leveraging Riemannian optimal transport and stochastic flow matching. We will evaluate and compare these models in the following sections.

Several benchmark papers have evaluated the introduction of state-of-the-art (SOTA) models in protein design. Scaffold-Lab Zheng et al. [2024] benchmarks generative models on unconditional generation and motif-scaffolding (24 motifs) in 4 dimensions: diversity, novelty, efficiency, and designability. MotifBench Zheng et al. [2025] uniquely focuses on motif-scaffolding task. Compared with Scafold-Lab, it extends the data size (30 motifs); considering the same evaluation dimension as Scaffold-Lab. FoldBench Xu et al. [2025] evaluates the monomer and interaction among proteins, nucleic acids, and ligands, with 9 tasks in total. However, their metric dimensions are simple, consisting only of DockQ, RMSD, and LDDT. Li et al. Li et al. [2025] use a variety of dimensions solely to benchmark unconditional generation and analyze the mathematics behind different architectures, but these types of mathematics are not related to the protein itself, just the theory of architecture. Prior evaluations provide useful baselines yet under-specify SE(3)-aware structure and perturbation stability; they compare models to each other, but do not seek to understand why they work well. This may help us choose the most suitable model for a specific task, but it gives us no indication of how to improve that model.

To address the above problems, this paper proposes FrameBench, a framework which specifically focuses on a special type of model: Frame-based models. **Our main contributions are:**

- We observe the theory both on structure representation (i.e., $SE(3)$ symmetry) and on architecture (SDE or ODE);
- In addition, we introduce new metrics to benchmark diversity and structure properties. Frame-based generators instantiate SE(3)-equivariant inductive bias; our evaluation mirrors that bias in the metrics.

2 Model Description

Formalization. State SE(3) action; define metrics (CATH entropy/coverage, SWORD2 partitions with VI); see Appendix C.

Implication beyond proteins. The same template applies to other generative domains with symmetries (e.g., planar SE(2), permutation groups, gauge symmetries), so this is a general recipe: (i)

identify symmetry; (ii) pick internal-geometry (or group-invariant) functionals; (iii) require invariance and smooth noise-stability; (iv) report compute-normalized efficiency. (details in Appendix E.)

The concept of *SE(3)* Murray et al. [2017], short for the Special Euclidean Group in three dimensions. Formally, $SE(3) = \{(R, t) \mid R \in SO(3), t \in \mathbb{R}^3\}$ represents the group of all rigid-body transformations combining rotations and translations. $SE(3)$ space is both a Riemannian Manifold and a Lie group.

Definition 1 (Riemannian Manifold)**.** *A* Riemannian manifold *is a smooth manifold $\mathcal{M}$ equipped with an inner product $\langle \cdot, \cdot \rangle_p$ on each tangent space $T_p\mathcal{M}$ that varies smoothly with the point $p \in \mathcal{M}$. The pair $(\mathcal{M}, \langle \cdot, \cdot \rangle)$ allows the measurement of lengths, angles, and geodesic distances on $\mathcal{M}$.*

Definition 2 (Lie Group)**.** *A* Lie group *$\mathcal{G}$ is a group that is also a smooth manifold such that the group operations — multiplication $(x, y) \mapsto xy$ and inversion $x \mapsto x^{-1}$ — are smooth maps. The tangent space at the identity element, denoted $\mathfrak{g} = T_e\mathcal{G}$, is called the* Lie algebra *of $\mathcal{G}$.*

Definition 3 (Special Orthogonal Group)**.** *We define the 3D spacial orthogonal group as the Lie group $SO(3)$ of origin preserving rotations on three dimensional Euclidean space. We can represent $SO(3)$ with 3×3 orthogonal matrices with determinant 1.*

Definition 4 (Special Euclidean Group $SE(3)$)**.** *The* Special Euclidean Group in three dimensions*, denoted $SE(3)$, is defined as*

$$SE(3) = \{(R, t) \mid R \in SO(3),\ t \in \mathbb{R}^3\},$$

where $SO(3)$ is the group of 3×3 rotation matrices satisfying $R^\top R = I$ and $\det(R) = 1$. The group operation corresponds to rigid-body composition:

$$(R_1, t_1) \circ (R_2, t_2) = (R_1R_2,\ R_1t_2 + t_1).$$

$SE(3)$ is a six-dimensional Riemannian Lie group *representing all rotations and translations in $\mathbb{R}^3$.*

This mathematical structure has long been used to describe spatial configurations and motions of physical systems. However, its integration into deep learning for biomolecular modeling was popularized by AlphaFold2 Bryant et al. [2022]. Specifically, AlphaFold2 introduced the Invariant Point Attention (IPA) mechanism, an *SE(3)*-equivariant architecture that ensures predictions remain consistent under global rotations and translations. This design allows the model to reason about protein geometry in a coordinate-free manner, learning physically meaningful spatial relationships. Following AlphaFold2, many generative models such as FrameDiff Yim et al. [2023b], FrameFlow Yim et al. [2023a], and FoldFlow Bose et al. [2023] explicitly adopt *SE(3)*-equivariant neural networks for protein backbone generation. These three models, due to their architectural similarity based on diffusion modeling and flow matching paradigms, constitute the primary focus of our investigation in this work.

Protein structures are made up of amino acids; each amino acid has $N - C_\alpha - C - O$ as nodes in 3D space. FrameDiff Yim et al. [2023b] uses the Gram-Schmidt process to transfer the 3D space representation to $SE(3)$ representation, then uses a diffusion model to generate the new elements in $SE(3)$ space. In each step of the diffusion model, they leverage IPA Bryant et al. [2022] to keep $SE(3)$ symmetry. This architecture is called FramePred; in other words, FramePred is an $SE(3)$-equivariant neural network that predicts denoised protein backbone frames and torsion angles from noisy inputs during the diffusion process. FrameDiff can use both the Euler-Maruyama integrator Higham [2001] for SDE sampling and the Euler integrator for ODE sampling.

FrameFlow Yim et al. [2023a] inherits the FramePred in FrameDiff, and also inherits the dataset for training. FrameDiff uses a diffusion model, which learns a score function; FrameFlow uses a flow matching model, which learns vector fields. As stated in their paper, FrameFlow achieves five times fewer sampling timesteps while achieving two times better designability compared to FrameDiff. But this result is just for timesteps, not directly comparing the forward pass time during training or for a single model inference.

FoldFlow Bose et al. [2023] includes: FoldFlow-base, a simulation-free deterministic flow matching model on $SE(3)$ that learns continuous-time dynamics using geodesic interpolants and closed-form conditional vector fields; FoldFlow-OT, accelerates training with Riemannian optimal transport to construct straighter and more stable probability paths between source and target distributions on $SE(3)$; FoldFlow-SFM, Learns stochastic dynamics via simulation-free Brownian bridges on $SE(3)$, enhancing robustness and novelty in high-dimensional protein backbone generation.

3 Benchmark

We evaluate the above 3 models in 3 dimensions: Efficiency, diversity and structural property. See the following subsections for details.

3.1 Efficiency

We evaluated the efficiency of our approach during both the training and inference phases. All experiments were performed on a workstation equipped with an NVIDIA RTX A6000 GPU (48 GB VRAM) running CUDA 13.0 and driver version 580.65.06.

For training, we used a dataset downloaded from the PDB on October 8, 2025, which was filtered to include proteins with lengths between 60 and 512 amino acids and a resolution better than 5Å. Run DSSP Kabsch and Sander [1983], then removed monomers with more than 50% loops. This filtering criterion is consistent with FrameDiff Yim et al. [2023b]. The final training set comprised 242,838 proteins.

Considering that FrameDiff can generate proteins up to a length of 500, FoldFlow can generate proteins up to a length of 300. For inference, we generated a total of 800 novel protein structures using the trained models, comprising 200 structures each at lengths of 100, 200, 300, and 500 amino acids. We set the time step for all models to 100; the number of parameters and inference time are shown in the following table.

Table 1: Calculation time (unit: seconds) for different protein lengths and different models. Each model generates 200 proteins per length.

Protein length (amino acids)	FrameDiff	FrameFlow	FoldFlow-base	FoldFlow-OT	FoldFlow-SFM
Length 100	3733.91	563.12	27149.87	9341.23	13524.73
Length 200	5547.69	917.81	57458.70	27967.43	19116.53
Length 300	8063.28	1489.70	59565.52	41412.42	31513.81
Length 500	15583.33	3009.82	82311.44	87979.40	58128.20

3.2 Diversity based on CATH

To evaluate diversity, we cluster the generated proteins using CATH Orengo et al. [1997]. CATH (Class-Architecture-Topology-Homology) is a well-recognized protein structure classification database that employs a hierarchical system to systematically classify protein structural domains. This classification framework comprises four main levels: Class, Architecture, Topology, and Homology, providing a systematic framework for understanding protein structural evolutionary relationships and functional diversity. We provide a comprehensive protein structure diversity assessment system that systematically analyzes generated protein structures based on the CATH classification framework, quantifies structural diversity, and diagnoses generation model biases. This tool is particularly valuable for evaluating and comparing different protein generation models, as well as understanding their structural coverage within the CATH classification space.

The generated backbones span multiple CATH classes, indicating broad architectural coverage. The assessment summary consolidates these signals, reporting high class diversity, a solid success rate, a moderate mean helix content, and the detection of numerous distinct CATH classes. See Figure 1. This cluster is just based on Class; furthermore, we try the cluster based on Architecture, the results can be seen in section A in the Appendix.

3.3 Structural properties based on SWORD2

For evaluating the structural properties, we use the webserver SWift and Optimized Recognition of protein Domains 2 (SWORD2, Cretin et al. [2022]), which partitioning algorithm produces multiple alternative domain assignments for given protein structures. The demo code for SWORD2 analysis of the domains of generated proteins is in section B of the Appendix.

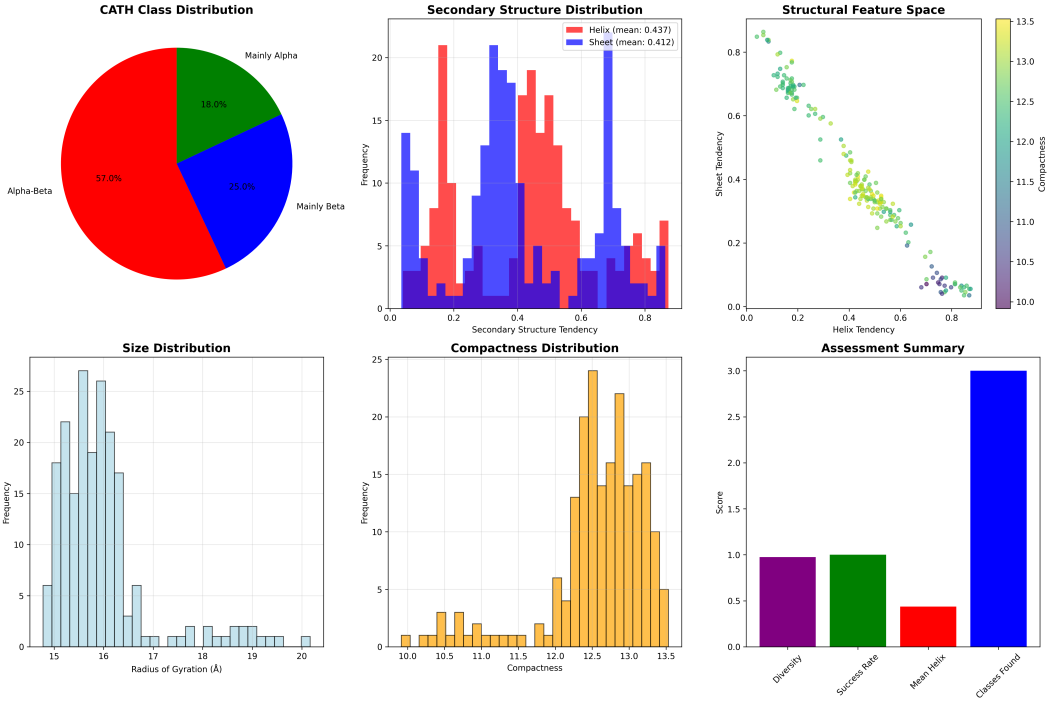

Figure 1: Visualization for the results of FrameDiff, protein length 200. (a) Distribution of proteins among the four major CATH classes. (b) Ten most frequent architectures identified across the dataset. (c) Class–architecture relationship matrix showing how architectures distribute across structural classes. (d) Feature space projection illustrating the relationship between helix content, sheet content, and overall compactness. (e) Average architecture detection scores indicating the degree of similarity to known structural motifs. (f) Summary of structural diversity across classes and architectures. **Take-home message:** The generated proteins display broad coverage across CATH classes and architectures, demonstrating that the model captures a wide range of fold topologies consistent with natural protein diversity rather than overfitting to specific structural families.

4 Discussion

Our results suggest that conventional scores, while informative, are incomplete for symmetry-structured generators; the proposed framework fills this gap.
In conclusion, this work makes a two-fold contribution to the field of AI-generated protein structures. First, it introduces a novel set of evaluation metrics, moving beyond conventional measures to provide a more nuanced and comprehensive assessment framework for the scientific community. Second, through extensive theoretical analysis, it offers profound insights into a class of generative models, elucidating their principles both in terms of structural representation and fundamental generative mechanisms. This dual advancement not only enriches the evaluative toolkit but also deepens our conceptual understanding of how these models operate.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Patrick Bryant, Gabriele Pozzati, and Arne Elofsson. Improved prediction of protein-protein interactions using alphafold2. *Nature communications*, 13(1):1265, 2022.

Gabriel Cretin, Tatiana Galochkina, Yann Vander Meersche, Alexandre G de Brevern, Guillaume Postic, and Jean-Christophe Gelly. Sword2: hierarchical analysis of protein 3d structures. *Nucleic acids research*, 50(W1):W732–W738, 2022.

Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

Wenran Li, Xavier Cadet, Cédric Damour, Yu Li, Alexandre de Brevern, Alain Miranville, and Frederic Cadet. Benchmark of diffusion and flow matching models for unconditional protein structure design. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Saber Saharkhiz, Mehrnaz Mostafavi, Amin Birashk, Shiva Karimian, Shayan Khalilollah, Sohrab Jaferian, Yalda Yazdani, Iraj Alipourfard, Yun Suk Huh, Marzieh Ramezani Farani, et al. The state-of-the-art overview to application of deep learning in accurate protein design and structure prediction. *Topics in Current Chemistry*, 382(3):23, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Sheng Xu, Qiantai Feng, Lifeng Qiao, Hao Wu, Tao Shen, Yu Cheng, Shuangjia Zheng, and Siqi Sun. Foldbench: An all-atom benchmark for biomolecular structure prediction. *bioRxiv*, pages 2025–05, 2025.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a.

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023b.

Zhuoqi Zheng, Bo Zhang, Bozitao Zhong, Kexin Liu, Zhengxin Li, Junjie Zhu, Jinyu Yu, Ting Wei, and Hai-Feng Chen. Scaffold-lab: Critical evaluation and ranking of protein backbone generation methods in a unified framework. *bioRxiv*, pages 2024–02, 2024.

Zhuoqi Zheng, Bo Zhang, Kieran Didi, Kevin K Yang, Jason Yim, Joseph L Watson, Hai-Feng Chen, and Brian L Trippe. Motifbench: A standardized protein design benchmark for motif-scaffolding problems. *arXiv preprint arXiv:2502.12479*, 2025.

5 Acknowledgements and Disclosure of Funding

WL is supported by a PhD grant from the Region Reunion and European Union (FEDER-FSE 2021/2027) 2023062, 345879. PEACCEL was supported through a research program partially co-funded by the European Union (UE) and Region Reunion (FEDER).

6 Disclaimer on Software Usage

In this study, some benchmarking experiments relied on third-party tools distributed under Non-Commercial (NC) licenses. These NC-licensed tools were executed exclusively by the academic collaborators in a non-commercial research setting. The industrial partner did not execute or run any NC-licensed code. Its role was limited to conceptual contributions. Accordingly, the use of NC-licensed software in this work remained strictly within an academic and non-commercial context, in compliance with the original license terms.

Appendix

Appendix Outline

- Section A Diversity based on CATH architecture and class distributions for proteins generated by FrameFlow.
- Section B Example of Python script demonstrating the use of SWORD2 for structural domain analysis.
- Section C Theoretical analysis of $SE(3)$ invariance, including assumptions, propositions, and proofs.
- Section D Formal definitions and implementation details of diversity, stability, and efficiency metrics.
- Section E Experimental framework for noise-stability evaluation, including noise models, statistics, and reproducibility settings.
- Section F List of abbreviations used throughout the paper.

A Diversity based on architecture

Using FrameFlow, protein length 200 as an example. See Figure 2.

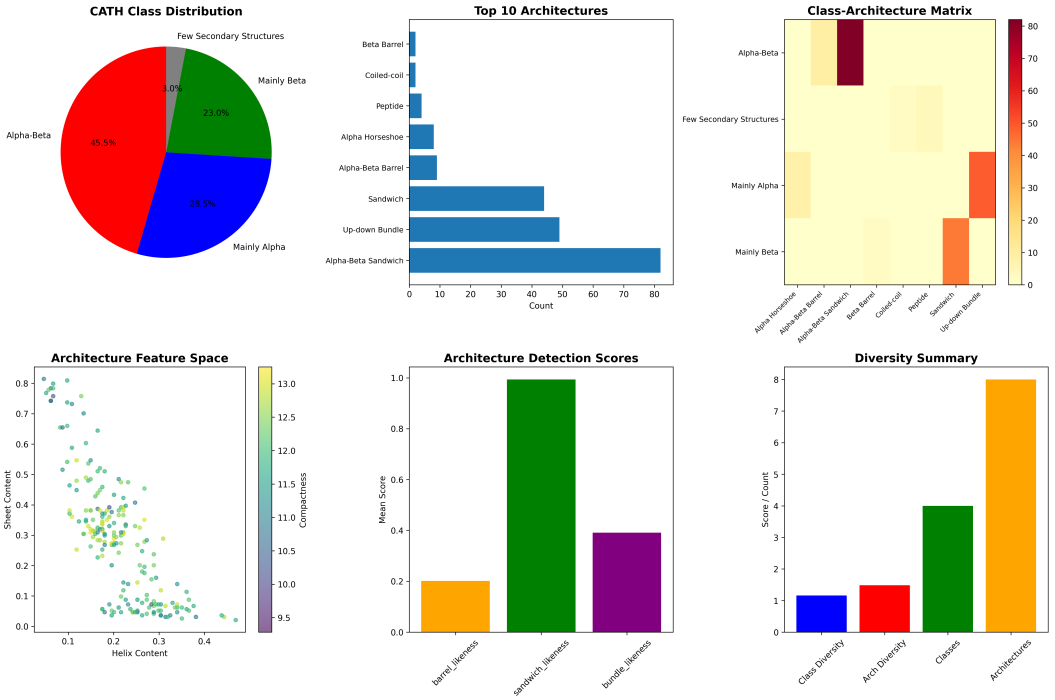

Figure 2: Diversity of 200 proteins generated by FrameFlow, protein length 200. This cluster is based on both Class and Architecture. (Top left) Distribution of proteins among the four major CATH classes. (Top middle) Top 10 most frequent architectures. (Top right) Class–architecture relationship matrix showing how architectures distribute across structural classes. (Bottom left) Feature space projection showing the relationships between helix content, sheet content, and structural compactness. (Bottom middle) Average architecture detection scores, indicating the degree of similarity to known structural motifs (e.g., barrel-, sandwich-, or bundle-like). (Bottom right) Summary of structural diversity across classes, architectures, and overall composition. **Take-home message:** FrameFlow generates structurally diverse and compositionally balanced proteins spanning multiple CATH classes and architectures, demonstrating its ability to explore a wide conformational landscape rather than overfitting to a specific fold type.

B Demo code to use SWORD2

```python
import subprocess
from concurrent.futures import ThreadPoolExecutor

# List of 5 PDB codes
pdb_list = ['1jx4', '2c78', '1f5n', '1a8y', '1b89']

# Function to run a SWORD2 job
def run_sword2(pdb_code):
    command = f'./SWORD2.py -p {pdb_code} -o results'
    try:
        subprocess.run(command, shell=True, check=True)
        print(f"Job for {pdb_code} completed successfully.")
    except subprocess.CalledProcessError as e:
        print(f"Error running job for {pdb_code}: {e}")

# Run 5 jobs using ThreadPoolExecutor
def run_sword2_jobs():
    with ThreadPoolExecutor(max_workers=5) as executor:
        executor.map(run_sword2, pdb_list)

if __name__ == "__main__":
    run_sword2_jobs()
```

C $SE(3)$ Invariance: Statements and Proofs

Assumption C.1 (Internal-geometry dependence)**.** *All structure functions used by our evaluation (domain partitioner π, class labeler c and any derived metrics) depend only on internal geometry of F (pairwise distances, relative orientations, and/or torsions), and not on its absolute pose in $\mathbb{R}^3$.*

C.1 Metric definitions (for reference)

Diversity. CATH entropy $H(\mathcal{B}) = -\sum_c P_{\text{CATH}}(c) \log P_{\text{CATH}}(c)$ and coverage $\text{cov}(\mathcal{B}) = \frac{\#\{c:\, P_{\text{CATH}}(c)>0\}}{\#\text{CATH classes}}$.

Domain stability. For two partitions π, π' of $\{1, \ldots, n\}$, the variation of information (VI) is

$$\text{VI}(\pi, \pi') \;=\; H(\pi) + H(\pi') - 2I(\pi, \pi'),$$

with entropies and mutual information computed over the node index set. We also record $\Delta\#$domains and a size-pattern shift (e.g., ℓ_1 distance between normalized domain-size histograms).

C.2 Core invariance claims

Proposition C.1 (Internal-geometry invariance)**.** *For any $g \in SE(3)$ and any indices i, j,*

$$d_{ij}(g{\cdot}F) = d_{ij}(F), \qquad \Delta R_{ij}(g{\cdot}F) = \Delta R_{ij}(F),$$

and torsions computed from F are unchanged by g. Hence any functional depending only on $\{d_{ij}, \Delta R_{ij}, \phi_i, \psi_i\}$ is $SE(3)$-invariant.

Proof. For d_{ij}, $\|Qp_i + t - (Qp_j + t)\| = \|Q(p_i - p_j)\| = \|p_i - p_j\|$ since Q is orthogonal. For ΔR_{ij}, $(QR_i)^\top (QR_j) = R_i^\top Q^\top Q R_j = R_i^\top R_j$. Standard torsions (e.g., Ramachandran) depend on internal dihedrals and are invariant to global pose. □

Proposition C.2 (Domain partition invariance)**.** *If π satisfies Assumption C.1, then $\pi(g{\cdot}F) = \pi(F)$ for all $g \in SE(3)$. Consequently,*

$$\text{VI}\big(\pi(F), \pi(g{\cdot}F)\big) = 0, \qquad \Delta\#\textit{domains} = 0, \quad \textit{and size-pattern shift } = 0.$$

Proof. By Assumption C.1, π is a function of internal geometry only; by Proposition C.1, internal geometry is unchanged by g. Thus, the partition is identical. Identical partitions have $\text{VI} = 0$, equal domain counts, and identical size distributions. □

Proposition C.3 (CATH-based diversity invariance)**.** *Let $\mathcal{B}$ be a batch of structures and $g \in SE(3)$. If the class labeler c satisfies Assumption C.1, then*

$$P_{CATH}(\mathcal{B}) \;=\; P_{CATH}(g\cdot\mathcal{B}), \quad H(\mathcal{B}) \;=\; H(g\cdot\mathcal{B}), \quad \mathrm{cov}(\mathcal{B}) \;=\; \mathrm{cov}(g\cdot\mathcal{B}).$$

Proof. Each $c(F)$ equals $c(g\cdot F)$ by Assumption C.1 and Proposition C.1. Therefore, the empirical distribution over classes is unchanged, and so are entropy and coverage. □

Corollary C.1 (Batch-level metric invariance)**.** *Any batch-aggregated functional of $c(F)$ and $\pi(F)$ that depends only on internal geometry (e.g., the metrics above) is invariant under simultaneous rigid motions of all structures in the batch.*

C.3 Noise stability: smooth variation under small coordinate σ-jitter

Beyond invariance to rigid motions, we stress *smooth degradation* of structure metrics under small, zero-mean coordinate noise.

Assumption C.2 (Centered Gaussian σ-jitter)**.** *Let J_σ map F to F' by $p'_i = p_i - \bar{p} + \sigma\varepsilon_i$ with $\bar{p} = \frac{1}{n}\sum_j p_j$ and $\varepsilon_i \sim \mathcal{N}(0, I_3)$; leave R_i unchanged (or re-orthogonalize if needed). This removes global translation and prevents spurious rigid motions.*

Proposition C.4 (Lipschitz-type stability of partitions)**.** *Suppose the partitioner π is locally Lipschitz w.r.t. internal distances and angles. Then there exists $\sigma_0 > 0$ and $L > 0$ such that for $0 < \sigma \le \sigma_0$,*

$$\mathbb{E}\big[\,\mathrm{VI}\big(\pi(F), \pi(J_\sigma(F))\big)\,\big] \;\le\; L\,\sigma\,.$$

Practical implication. Small σ should yield near-zero median VI with narrow IQR; the VI–σ curve should increase smoothly (no abrupt jumps) unless the structure lies near a partition boundary.

C.4 Worked example (what to expect)

On random $g \in SE(3)$ we expect exact invariance:

$$\mathrm{VI}\big(\pi(F), \pi(g\cdot F)\big) = 0, \qquad H(\mathcal{B}) = H(g\cdot\mathcal{B}), \qquad \mathrm{cov}(\mathcal{B}) = \mathrm{cov}(g\cdot\mathcal{B}).$$

Under small σ, VI should remain near zero and grow smoothly, supporting the stability claim in Proposition C.4.

C.5 Notes on implementation

(i) Use a fixed random seed per structure for g and for jitter; (ii) re-orthogonalize R_i after any numeric perturbation; (iii) when a downstream tool introduces tie-breaking randomness, average over repeated runs or fix its seed; (iv) declare numerical tolerances (e.g., treat $\mathrm{VI} < 10^{-9}$ as zero).

D Metric Definitions and Implementation Details

D.1 Diversity (CATH) Metrics

Definition 5 (Empirical CATH distribution)**.** *Let $\mathcal{B} = \{F^{(1)}, \ldots, F^{(N)}\}$ be a batch of generated structures and $c(F) \in \mathcal{C}$ the CATH label (at a chosen hierarchy level, e.g.,* Class *or* Architecture*). The empirical distribution is*

$$p_c \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\, c(F^{(i)}) = c\,\}, \qquad c \in \mathcal{C}.$$

Definition 6 (Entropy, effective classes, coverage)**.** *With natural logarithms,*

$$H(\mathcal{B}) = -\sum_{c\in\mathcal{C}} p_c \log p_c, \quad N_{\mathrm{eff}}(\mathcal{B}) = \exp\big(H(\mathcal{B})\big), \quad \mathrm{cov}(\mathcal{B}) = \frac{|\{c :\, p_c > 0\}|}{|\mathcal{C}|}.$$

Calibration vs. training distribution (optional). If q_c denotes the training distribution on the same CATH level,

$$D_{\mathrm{KL}}(p\|q) = \sum_c p_c \log \frac{p_c}{q_c}, \qquad \mathrm{TV}(p,q) = \tfrac{1}{2}\sum_c |p_c - q_c|, \qquad \mathrm{JSD}(p\|q) = \tfrac{1}{2}D_{\mathrm{KL}}(p\|m) + \tfrac{1}{2}D_{\mathrm{KL}}(q\|m),$$

with $m = \frac{1}{2}(p+q)$. Report at most one divergence (e.g., JSD) to avoid redundancy.

SE(3) invariance. If the labeler c depends only on internal geometry (Assumption C.1), then H, N_{eff}, and cov are invariant to rigid motions (Proposition C.3).

Recommended reporting. Provide H, N_{eff}, and cov with 95% bootstrap CIs (resampling over structures). State the CATH level used and how q (train) was estimated if calibration is reported.

D.2 Domain-Structure Stability (SWORD2) Metrics

Definition 7 (Partitions and block probabilities)**.** *A domain partition π of $\{1,\dots,n\}$ is a set of disjoint blocks $\{B_k\}_{k=1}^{K}$ with $\bigcup_k B_k = \{1,\dots,n\}$. Let $p_k = |B_k|/n$ and, for two partitions $\pi = \{B_k\}$ and $\pi' = \{B'_\ell\}$,*

$$p_{k\ell} = \frac{|B_k \cap B'_\ell|}{n}, \qquad p'_\ell = \sum_k p_{k\ell}.$$

Definition 8 (Variation of Information (VI))**.** *The VI between π and π' is*

$$\mathrm{VI}(\pi,\pi') \;=\; H(\pi)+H(\pi')-2I(\pi,\pi') \;=\; \Big(-\sum_k p_k \log p_k\Big)+\Big(-\sum_\ell p'_\ell \log p'_\ell\Big)-2\sum_{k,\ell} p_{k\ell}\log\frac{p_{k\ell}}{p_k\,p'_\ell}.$$

Auxiliary stability indicators.

$$\Delta\#\mathrm{domains} \;=\; |K-K'|, \qquad \Delta\text{size-pattern} \;=\; \tfrac{1}{2}\sum_s \big|h_\pi(s) - h_{\pi'}(s)\big|,$$

where h_π is a histogram of normalized domain sizes (choose a fixed binning over $(0,1]$).

SE(3) invariance and noise stability. If π depends only on internal geometry, then $\mathrm{VI}(\pi(F), \pi(g\cdot F)) = 0$ for any $g \in SE(3)$ (Prop. C.2). Under small jitter J_σ (Assumption C.2), VI should grow smoothly with σ (Prop. C.4).

Tolerance rates and curves. For a threshold $\tau > 0$,

$$T(\sigma;\tau) \;=\; \Pr_{F\sim\mathcal{B}}\left[\mathrm{VI}(\pi(F), \pi(J_\sigma(F))) \le \tau\right],$$

reported with 95% bootstrap CIs over structures. Also report the median and IQR of VI as a function of σ.

Recommended reporting. (1) Exact SWORD2 version/flags; (2) $T(\sigma;\tau)$ for a fixed τ (e.g., 0.05 or 0.10); (3) $\Delta\#$domains and Δsize-pattern summaries.

D.3 Efficiency (Compute-Normalized) Metrics

Definition 9 (Sampling throughput and acceptance)**.** *For a model M, length ℓ, and hardware h, let N_{samp} be the number of attempted samples in wall-clock time t, and N_{acc} the number passing a predefined structural filter (e.g., geometry validity). Define the throughput $R_{samp} = N_{samp}/t$ and acceptance $\alpha = N_{acc}/N_{samp}$.*

Fair comparison protocol. (i) Report results at *matched lengths* across models; (ii) if some models cannot sample certain lengths (e.g., $\ell = 500$), separate the analysis; (iii) fix hardware, precision, and compiler flags; (iv) use identical structural filters for N_{acc}.

Recommended reporting. Table per length with columns: *throughput* R_{samp}, *acceptance* α, Eff, and Eff_{norm}, plus the definition (or estimate) of $C(M, \ell)$ or $\widehat{C}(M, \ell)$.

E Noise-Stability Protocol

This appendix operationalizes the *noise-stability* checks introduced in the main text: how to inject controlled coordinate noise, which σ-values to test, which statistics to report, and how to ensure results are reproducible and comparable across models and lengths.

E.1 Noise models

Rotations R_i are kept as-is (re-orthogonalize via polar decomposition if perturbed numerically). Centering removes spurious translation; the partitioner π is $SE(3)$-invariant, so downstream quantities are unaffected by any global pose drift.

Torsional jitter (optional, physically informed). As a sensitivity analysis, one may perturb backbone torsions (ϕ, ψ, ω) with small zero-mean Gaussian noise and rebuild coordinates by forward kinematics, then re-apply the same stability pipeline. Report which noise family is used.

E.2 Interpreting σ (units and calibration)

Coordinates are in Å. For Cartesian jitter, the expected per-atom displacement is $\sqrt{\mathbb{E}\|\sigma\varepsilon\|^2} = \sqrt{3}\,\sigma$; hence a target RMSD ρ corresponds to $\sigma \approx \rho/\sqrt{3}$. This lets readers map σ to a familiar scale.

E.3 Recommended σ-grid and repetitions

Use a small grid covering the regime of "barely noticeable" to "moderate" perturbations, e.g.

$$\sigma \in \{0,\ 0.05,\ 0.10,\ 0.20,\ 0.30,\ 0.50\}\ \text{Å}.$$

For each $\sigma > 0$, perform M independent jitters per structure (default $M = 3$) and average the resulting statistics for that structure to reduce Monte Carlo variance.

E.4 Primary statistics to report

Let π be the domain partitioner. For each structure F and each σ:

- **VI–σ curve:** median and IQR of $\text{VI}(\pi(F), \pi(J_\sigma(F)))$ over the batch.
- **Tolerance rate** at threshold τ: $T(\sigma; \tau) = \Pr[\text{VI} \le \tau]$ with 95% bootstrap CIs (resampling structures).
- **Auxiliary shifts:** median (IQR) of $\Delta\#$domains and size-pattern shift (Section D).

Recommended thresholds: $\tau \in \{0.05,\ 0.10\}$.

E.5 Monotonicity and smoothness diagnostics

Noise-stability should *increase smoothly* with σ (Proposition C.4). To summarize smoothness, report (i) the finite-difference slope at the origin $s_0 = \frac{\text{median}(\text{VI}_{\sigma_2}) - \text{median}(\text{VI}_{\sigma_1})}{\sigma_2 - \sigma_1}$ for the two smallest nonzero grid points, and (ii) the maximum one-step jump $\max_j \left|\,\text{median}(\text{VI}_{\sigma_{j+1}}) - \text{median}(\text{VI}_{\sigma_j})\right|$. Large jumps indicate proximity to decision boundaries in π.

E.6 Reproducibility settings

Fix a seed per structure for the jitter draws and any downstream randomness. Cache the raw partitions returned by π to avoid version drift. Declare numerical tolerances and report the SWORD2 (version, flags, threading) used.

F List of Abbreviations

AA Amino Acid
AF2 AlphaFold2
CATH Class–Architecture–Topology–Homologous superfamily (protein structure classification)
ESM Evolutionary Scale Modeling
ESMFold ESM-based single-sequence protein folding model
FM Flow Matching
FrameBench Symmetry-aware evaluation framework for frame-based protein generative models
FrameDiff Frame-based Diffusion Model
FrameFlow Frame-based Flow Matching Model
FoldFlow-OT FoldFlow with Optimal Transport path construction
FoldFlow-SFM FoldFlow with Stochastic Flow Matching
GPU Graphics Processing Unit
MSE Mean Squared Error
OT Optimal Transport
PDB Protein Data Bank
RMSD Root Mean Square Deviation
SDE Stochastic Differential Equation
SE(3) Special Euclidean group in 3D (rigid-body transformations)
SFM Stochastic Flow Matching
SO(3) Special Orthogonal group in 3D (rotations)
TM-score Template Modeling Score
VI Variation of Information (clustering/partition distance)