TEMPORAL TRIANGULAR ATTENTION FOR PROTEIN MOLECULAR DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein dynamics carry critical information about protein function, yet current machine learning methods offer few ways to encode and reason over them. Molecular dynamics (MD) trajectories provide a rich view of these dynamics that is complementary to the sequence and evolutionary signals exploited by protein language models, but transforming them into effective protein representations remains an open challenge. In this work, we introduce *Temporal Triangular Attention* (TTA), which extends triangular attention from a single static structure to a temporally evolving sequence of structures. For each residue pair, TTA jointly selects a mediating residue and a trajectory frame, enabling pair-conditioned reasoning over when and where relevant motion occurs. We instantiate TTA in ProTTA, an Evoformer-style MD encoder that incorporates residue trajectories into pairwise structural embeddings. By pretraining ProTTA to forecast the dynamic cross-correlation of the next trajectory window, we obtain representations of coordinated motion without any functional annotations. We evaluate ProTTA on two benchmarks spanning function prediction and dynamics forecasting. For function prediction, ProTTA outperforms existing MD-based encoders and, when combined with sequence information, achieves the best result on 22 of 24 metrics against both sequence models and MD encoders. For dynamics forecasting, ProTTA outperforms the MD-only baseline and performs comparably to a sequence-conditioned model while using dynamics alone.

1 INTRODUCTION

Understanding how proteins carry out their biological functions has long been central to biology. Computational biology has translated this to a protein function prediction task applying various approaches(Radivojac et al., 2013; Zhou et al., 2019; Yu et al., 2023). Modern methods infer annotations such as Gene Ontology (GO) terms (gen, 2026) from combinations of amino-acid sequence, evolutionary couplings, and three-dimensional structure (Kulmanov et al., 2024; Jang et al., 2024; Wang et al., 2025). Among these methods, protein language models (pLMs) provide particularly strong sequence representations and achieve some of the state-of-the-art performance in predicting function (Elnaggar et al., 2021; Lin et al., 2023; Su et al., 2024). Trained on sequences, pLMs implicitly capture sequence similarity, residue contacts and evolutionary context in their embedding space, and structure-aware variants further incorporate explicit structural information (Su et al., 2024). Yet, these models generally represent a protein by its sequence, a single static structure, or a small unordered set of conformations.

Proteins, however, are not static. The ability of proteins to function as catalyst, receptor, and transportor are mediated by its motions and transitions between its functional conformations (Eisenmesser et al., 2005; Boehr et al., 2009). The dynamics of protein is commonly simulated by molecular dynamics (MD) simulations, which involves physics based simulation of atomic level interaction. MD trajectories can therefore provide dynamic properties of proteins that is not present in sequence or in a single structure. This suggests that a representation capturing protein motion could provide information complementary to sequence-based models and improve predictions of protein function.

Transforming an MD trajectory into a representation that is useful for function prediction remains an open challenge. Only a few existing methods combine pLMs with MD-derived information (Hou et al., 2026; Kalifa et al., 2026), and they typically handle dynamics by distilling it into sequence fea-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tures, pooling across conformations, or compressing the trajectory into a single embedding. While these approaches can capture temporal or ensemble-level signals, but do not support explicit, pair-conditioned reasoning over a trajectory. In particular, they discard the residue-pair representation where geometric structure models rely, which is a natural representational space to capture dynamics of proteins.

We address this gap by introducing *Temporal Triangular Attention* (TTA), an extension of the triangular attention mechanism used in AlphaFold-style structure models (Jumper et al., 2021; Ahdritz et al., 2024; Lin et al., 2023). In our proposed approach, a directed target edge $i \to j$ at frame t, TTA jointly attends over a third residue k and a frame u. It scores the supporting edges $(i, t) \to (k, u)$ and $(j, t) \to (k, u)$ conjunctively. Joint normalization over (k, u) lets each target pair select both *which residue* and *when* in the trajectory carries relevant information about movements. By placing this operation inside an Evoformer-style pair stack, TTA allows for encoding motion into the pair representation which is used for spatial reasoning. The standard triangle inequality is also applicable to the aligned cross-frame points and provides a geometric motivation for the proposed approach (Sec. B). We further evaluate how TTA serves a general MD embedding method that brings benefit beyond function prediction using forecasting task.

We instantiate TTA in ProTTA, an Evoformer-style encoder in which TTA replaces triangular self-attention after triangular multiplicative updates. The resulting pair representation captures pairwise interactions and movements of residues and is combined with pLM sequence embeddings for function prediction. We further pretrain ProTTA with a self-supervised objective that predicts the dynamic cross-correlation (DCC) matrix (Ichiye & Karplus, 1991; Hünenberger et al., 1995) of the next trajectory window. This initializes the pair representation to capture evolving coordinated motion before it is fine-tuned on function labels.

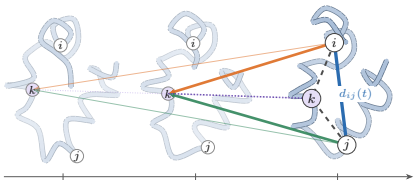

Figure 1: Illustration of temporal triangular attention (TTA). Each residue pair (i, j) is updated based on the motion traces of other residues k across frames.

We evaluate on mdCATH and ATLAS (Mirarchi et al., 2024; Vander Meersche et al., 2024) whether this dynamics representation adds information beyond sequence for function prediction. When fused with ESM-2, ProTTA improves over the sequence model on all 24 function-prediction metrics across mdCATH and ATLAS and ranks first on 22 of them, ahead of both structure-aware pLMs and prior MD encoders. The gains also concentrate where dynamics should matter most, like MF molecular function (13.9% in F_{max} on mdCATH). Beyond function prediction, ProTTA reduces forecasting RMSD by 20% relative to MD-only baselines and comes within reach of sequence-conditioned ones without access to sequence.

Our contributions are as follows:

- **Dynamics-aware representations for protein function prediction.** We show that representations learned from short-timescale MD trajectories improve both GO term and protein-level function prediction on two MD datasets.
- **Pair-conditioned temporal triangular attention.** TTA extends triangular attention from a single static structure to a temporally evolving sequence of structures. By jointly selecting a mediating residue and frame for each directed residue pair, it enables explicit reasoning over time while preserving pair-specific spatiotemporal dependencies.
- **Self-supervised pretraining and generality.** We pretrain ProTTA to forecast next-window DCC without functional annotations, and the resulting representations improve every function-prediction metric over ESM-2, outperform all evaluated MD encoders. We also show that ProTTA training from scratch can transfer to dynamics forecasting.

2 Related Works

2.1 Structure-aware Protein Embedding

Modeling protein structure has long been a central problem in computational biology, since a protein's structure and its conformational dynamics are closely tied to its function (Eisenmesser et al.,

2005; Boehr et al., 2009). Earlier methods obtain structure-aware embeddings by applying graph neural networks to residue-level graphs (Gligorijević et al., 2021; Lai & Xu, 2022; Zhang et al., 2022), thereby merging structural information into amino-acid sequence embeddings. More recent work scales up attention-based architectures. Protein language models (pLMs) (Lin et al., 2023; Su et al., 2024) learn residue embeddings through large-scale pretraining and capture structure either implicitly or through structure-aware tokens, while structure prediction models (Jumper et al., 2021; Ahdritz et al., 2024) maintain an explicit pairwise representation that is updated by triangular attention. Although these models achieve state-of-the-art performance on tasks such as protein function prediction, they represent each protein by a single static structure and therefore cannot take MD trajectories as input or capture conformational dynamics.

2.2 Learning from MD Trajectories

Most existing MD embedding models pool or aggregate the trajectory into a single state (Chiang et al., 2022; Wu et al., 2023; Xu et al., 2024), resembling static structure-aware embedding methods and consequently do not capture structural motion. A few recent approaches instead model MD trajectories directly. MDGen (Jing et al., 2024b) uses a flow-based generative model to capture relationships across frames, and ConfRover (Shen et al., 2025) encodes each frame with triangular attention before fusing the resulting frame embeddings through a causal late-fusion module. However, both are designed to generate trajectories rather than to embed them, and in ConfRover, temporal information is introduced only after each frame has been encoded independently. As a result, neither yields a representation learned directly from residue motion. We address this limitation with temporal-aware triangular attention (TTA), which makes the pairwise update itself aware of motion across frames.

3 Methodology

3.1 Preliminary: Triangular Mechanism

The triangular mechanism is the core operation in state-of-the-art protein structure prediction models. Originally proposed in AlphaFold2 (Jumper et al., 2021) and adopted by later models (Lin et al., 2023; Ahdritz et al., 2024; Abramson et al., 2024), it starts from the intuition that a 3D structure can be described by pairwise representations that is consistent under Euclidean geometry. The operations of triangle mechanism are over the pair representation given by $\mathbf{Z} \in \mathbb{R}^{L\times L\times c_z}$, where $\mathbf{z}_{ij}$ encodes the relationship or interaction between residues i and j of a length-L protein and is trained to be predictive of their distance d_{ij}. A necessary condition for consistency is the triangle inequality $d_{ij} \le d_{ik} + d_{jk}$, which couples the three edges of every triangle (i, j, k). The triangular mechanism encourages this consistency by updating each edge ij from the other two edges of every triangle, using two operators: triangular self-attention, and triangular multiplicative update. We provide detailed equations for both operations in Appendix A.

Below, $\mathrm{Linear}(\cdot)$ denotes a learned linear layer with its own parameters at each occurrence, LN denotes layer normalization, σ is the sigmoid function, and all projections of $\mathbf{z}_{ij}$ are applied after layer normalization.

3.2 Temporal Triangular Attention

A key limitation of triangular attention is that it operates on a single static structure: edge (i, j) is updated through third residues k of the same conformation, so the resulting representation cannot capture motion across conformations. We propose Temporal Triangular Attention (TTA), in which the pair representation is updated through the temporal traces of k and therefore encodes motion information.

Trace embedding. In addition to the pair representation, we maintain a trace representation that describes how each residue moves over time. We consider a window of S consecutive MD frames, superposed to remove global motion, and denote by $\mathbf{x}_i(t) \in \mathbb{R}^3$ the C_α coordinate of residue i at frame $t \in \{1, \dots, S\}$ and by $\bar{\mathbf{x}}_i = \frac{1}{S}\sum_t \mathbf{x}_i(t)$ its mean position over the window. Following the backbone-frame construction of AlphaFold2 (Jumper et al., 2021), we attach to each residue

a local frame $R_i \in \mathrm{SO}(3)$, whose three orthonormal axes are determined by the chain directions at residue i in the mean structure, here computed from consecutive C_α atoms. We then compute $\mathbf{y}_i(t) = R_i^\top \big(\mathbf{x}_i(t) - \bar{\mathbf{x}}_i\big) \in \mathbb{R}^3$, which expresses the displacement of residue i from its mean position in chain-relative coordinates. Since R_i rotates together with the protein, $\mathbf{y}_i(t)$ is invariant to global rotations and translations. The trace token is a linear embedding of this displacement plus a learned time embedding $\boldsymbol{\tau}_t$:

$$\mathbf{m}_i(t) = \mathrm{LN}\big(\mathrm{Linear}(\mathbf{y}_i(t)) + \boldsymbol{\tau}_t\big) \in \mathbb{R}^{c_m}. \tag{1}$$

Temporal triangular attention. Based on the trace representation, we replace triangular self-attention with a trace-aware update. As in vanilla triangular attention, for each residue pair (i, j) TTA seeks a third residue k, now one whose *motion* is related to that of both i and j, and updates $\mathbf{z}_{ij}$ accordingly. We score the two remaining sides of the triangle separately as two *legs*: the *start leg* measures the relationship (i, k) and the *end leg* measures (j, k). We do such a leg seperation since a score on the start leg alone would not depend on j, so every edge in row i would receive the same message, and the end leg makes the selection specific to the pair. We consider k important only if both legs score highly. While i and j are taken at the same frame t, k is read over its whole trace $u = 1, \ldots, S$, since coupled motions need not be simultaneous. All quantities below are per attention head of width c and we omit the head index.

Specifically, let $\mathbf{q}_i^{\mathrm{s}}(t) = W^{q,\mathrm{s}}\mathbf{m}_i(t)$ and $\mathbf{q}_j^{\mathrm{e}}(t) = W^{q,\mathrm{e}}\mathbf{m}_j(t)$ be the queries of i and j for the start and end legs, $\mathbf{k}_k^{\mathrm{s}}(u) = W^{k,\mathrm{s}}\mathbf{m}_k(u)$ and $\mathbf{k}_k^{\mathrm{e}}(u) = W^{k,\mathrm{e}}\mathbf{m}_k(u)$ the corresponding keys of k, and $\mathbf{v}_k(u) = W^v\mathbf{m}_k(u)$ its value. The pair representation contributes a bias $b_{ij} = \mathrm{Linear}(\mathbf{z}_{ij})$ and a gate $\mathbf{g}_{ij} = \sigma(\mathrm{Linear}(\mathbf{z}_{ij}))$, as in Eqs. 9–10. The two legs are scored by

$$\lambda_{ik}^{\mathrm{s}}(t, u) = \tfrac{1}{\sqrt{c}}\, \mathbf{q}_i^{\mathrm{s}}(t)^\top \mathbf{k}_k^{\mathrm{s}}(u) + b_{ik}, \qquad \lambda_{jk}^{\mathrm{e}}(t, u) = \tfrac{1}{\sqrt{c}}\, \mathbf{q}_j^{\mathrm{e}}(t)^\top \mathbf{k}_k^{\mathrm{e}}(u) + b_{jk}, \tag{2}$$

and combined conjunctively as a sum of log-sigmoids,

$$\ell_{ijk}(t, u) = \log \sigma\big(\lambda_{ik}^{\mathrm{s}}(t, u)\big) + \log \sigma\big(\lambda_{jk}^{\mathrm{e}}(t, u)\big), \tag{3}$$

which measures how compatible the motion of k at frame u is with the motions of i and j at frame t. Since the sum of logs is the log of a product, ℓ is high only if both legs are compatible. The scores are normalized jointly over third residues and frames with a softmax, selecting *which* residue to read and *when*:

$$\alpha_{ijk}(t, u) = \operatorname*{softmax}_{(k,u)}\ \ell_{ijk}(t, u). \tag{4}$$

The attended values are gated by the pair representation and concatenated over heads:

$$\mathbf{o}_{ij}(t) = \mathrm{concat}_{\text{heads}}\Big(\mathbf{g}_{ij} \odot \sum_{k=1}^{L}\sum_{u=1}^{S} \alpha_{ijk}(t, u)\, \mathbf{v}_k(u)\Big). \tag{5}$$

A key property of TTA is that it respects the triangle inequality. The vanilla triangular mechanism is designed around the triangle inequality, encouraging pairwise embeddings to remain geometrically consistent in Euclidean space. TTA extends this principle to Minkowski space, which allows the model to naturally capture structural motions across conformations (see Appendix B for theoretical support). As a result, TTA is beneficial not only for the protein-level objectives that motivated its design, but also for residue-level tasks such as structure forecasting, as we will demonstrate empirically in Section 4.4.

Empirical adaptations and approximations. The formulation above is expensive in two respects. Scoring every third residue for every edge and every pair of frames costs $O(S^2L^3)$, and a time-resolved pair representation updated by $\mathbf{o}_{ij}(t)$ at every frame would multiply the memory of $\mathbf{Z}$ by S, which is prohibitive for MD data with high sampling frequency.

We therefore make two adaptations. First, we restrict k to the spatial neighborhood of i, $k \in \mathcal{N}(i)$, the K residues nearest to i in the window-mean structure. The softmax in Eq. equation 4 and the sum in Eq. equation 5 then run over $\mathcal{N}(i)$ only, reducing the cost to $O(S^2L^2K)$. This restriction also motivates the separate projections of the two legs. The start leg always compares i with one of its spatial neighbours, whereas the end leg compares j with a residue at arbitrary distance. Second,

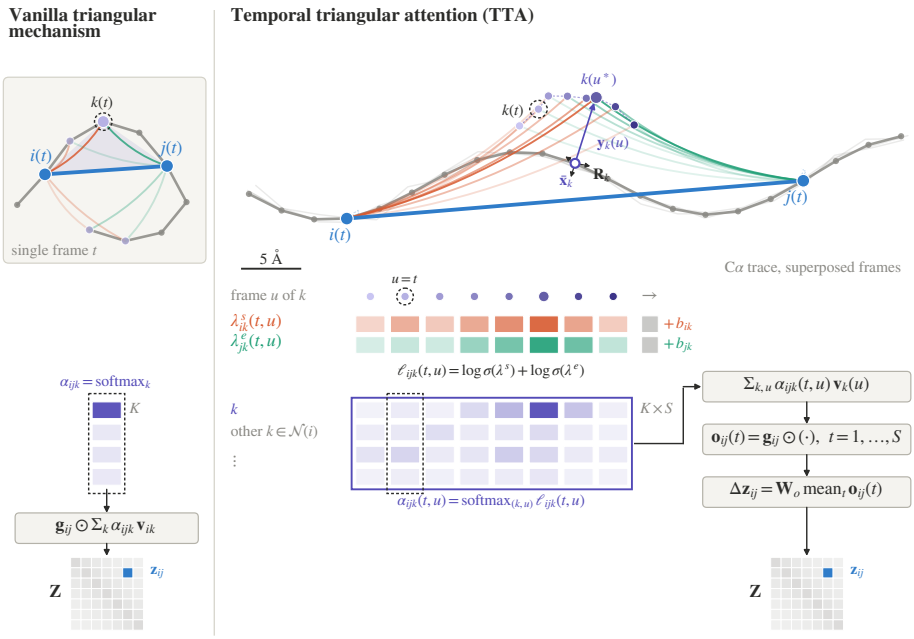

Figure 2: Illustration of the TTA computation. Unlike the vanilla triangular mechanism, which restricts each residue pair (i, j) to interact with third residues k within the same frame, TTA also searches over k in frames across the available window, allowing the pairwise embedding to be updated with information about structural motion.

we split each MD trajectory into windows of S frames and maintain a single pair representation per window, shared by all of its frames. The TTA update of a window is the average of its per-frame messages followed by an output projection:

$$\mathrm{TTA}(\mathbf{z}, \mathbf{m})_{ij} = W^o \frac{1}{S} \sum_{t=1}^{S} \mathbf{o}_{ij}(t). \tag{6}$$

3.3 ProTTA: Architecture and Pretraining

Building on TTA, we introduce ProTTA, an MD embedding model that follows the Evoformer pipeline but replaces triangular self-attention with TTA. Each of the blocks of ProTTA applies four residual updates:

$$\mathbf{z} \leftarrow \mathbf{z} + \mathrm{TriMul}_{\mathrm{out}}(\mathbf{z}), \quad \mathbf{z} \leftarrow \mathbf{z} + \mathrm{TriMul}_{\mathrm{in}}(\mathbf{z}),$$
$$\mathbf{z} \leftarrow \mathbf{z} + \mathrm{TTA}(\mathbf{z}, \mathbf{m}), \quad \mathbf{z} \leftarrow \mathbf{z} + \mathrm{Transition}(\mathbf{z}),$$

where $\mathrm{TriMul}_{\mathrm{out}}$ and $\mathrm{TriMul}_{\mathrm{in}}$ are the outgoing and incoming triangular multiplicative updates of Eq. equation 11, and $\mathrm{Transition}$ is a two-layer MLP. The trace tokens $\mathbf{m}$ are fixed inputs shared by all blocks and only the pair representation is updated, so that motion information accumulates in $\mathbf{Z}$ across blocks. Before the first block, $\mathbf{Z}$ is initialized from the windowed-mean structure, using embedded pairwise distances and relative sequence positions (details in Appendix C), so that it starts as a purely static description and all motion information enters through TTA. After the last block, we expect the output $\mathbf{Z}$ of ProTTA to be a pairwise embedding that jointly encodes the structure and its motion.

To initialize its parameters, we pretrain ProTTA to predict the pairwise dynamic cross-correlation (DCC) of residue motions. For a trajectory window $\mathcal{W}$,

$$\mathrm{DCC}_{ij}(\mathcal{W}) = \frac{\langle \Delta\mathbf{x}_i \cdot \Delta\mathbf{x}_j \rangle_{\mathcal{W}}}{\sqrt{\langle \|\Delta\mathbf{x}_i\|^2 \rangle_{\mathcal{W}} \langle \|\Delta\mathbf{x}_j\|^2 \rangle_{\mathcal{W}}}}, \qquad \Delta\mathbf{x}_i(t) = \mathbf{x}_i(t) - \langle \mathbf{x}_i \rangle_{\mathcal{W}}, \tag{7}$$

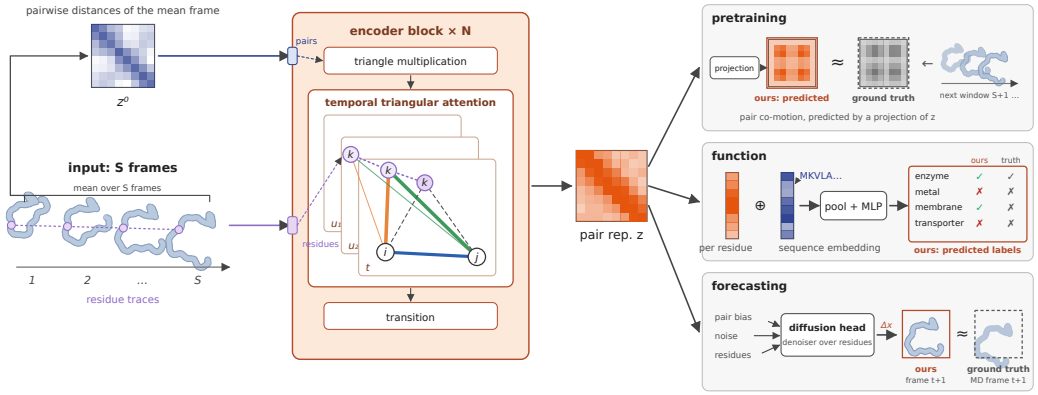

Figure 3: Architecture overview of ProTTA. Initialized by pairwise distances, the pair representation is fed through evoformer-inspired blocks which involves TTA calculation and triangular multiplicative updates. The output pairwise representation is then fed to downstream tasks.

where $\langle\cdot\rangle_{\mathcal{W}}$ denotes the average over the frames of $\mathcal{W}$. $\mathrm{DCC}_{ij} \in [-1, 1]$ is therefore positive when residues i and j move in the same direction and negative when they move in opposite directions.

Given window $\mathcal{W}_n$ as input, we optimize ProTTA to output the DCC of the *next* window $\mathcal{W}_{n+1}$ from its final pair representation through a symmetrized linear head, $\widehat{\mathrm{DCC}}_{ij} = \frac{1}{2}\,\mathbf{w}^\top(\mathbf{z}_{ij} + \mathbf{z}_{ji})$, and train it with the pearson loss:

$$\mathcal{L} = 1 - \mathrm{Pearson}_{i\neq j}\left(\widehat{\mathrm{DCC}}_{ij},\ \mathrm{DCC}_{ij}(\mathcal{W}_{n+1})\right). \tag{8}$$

An overview of ProTTA is available in Figure 3.

4 Experiments

4.1 Datasets and Downstream Tasks

We evaluate on two benchmarks, mdCATH (Mirarchi et al., 2024) and ATLAS (Vander Meersche et al., 2024), both curated protein databases that provide molecular dynamics (MD) trajectories. For ATLAS, we use the full MD trajectories and downsample them with a stride of 100 ps. For mdCATH, whose trajectories vary in simulation length, we use the full trajectories and maintain a uniform stride of 1 ns. Statistics of datasets are shown in Appendix C.1.

We consider two downstream tasks: protein function prediction and dynamics forecasting.

Function Prediction This task measures how much the MD information encoded by ProTTA benefits the prediction of protein function labels. We extract raw labels from UniProt protein database (uni, 2025) and further adopt two labeling schemes: (1) Gene Ontology (GO) labels (gen, 2026), which comprise fine-grained function terms organized under three root categories: molecular function (MF), biological process (BP), and cellular component (CC), and (2) 4-label human protein atlas (HPA) function labels (Uhlén et al., 2015), which are high-level protein function annotations established referring to the HPA database. Labeling procedure are detailed in Appendix C.1.

Dynamics Forecasting To demonstrate the utility of TTA as an MD embedding model beyond function prediction, we evaluate ProTTA on dynamics forecasting, which measures how well a model predicts the future evolution of an MD trajectory from a given initial structure. Following ConfRover (Shen et al., 2025), the model is conditioned on a single starting structure and generates the subsequent 100 ns of the trajectory in a rolling manner. Specifically, at the beginning of rolling, we limit the window size of TTA since the inputs are shorter than one window. We therefore did not load pretrained checkpoints and train the model from scratch.

4.2 BASELINES

We compare against three categories of baselines: two for function prediction and one for dynamics forecasting.

Function Prediction Baselines We include state-of-the-art sequence-based models for function prediction: ESM-2 (Lin et al., 2023), SaProt (Su et al., 2024), SeqDance (Hou et al., 2026). Comparing ProTTA against these models tests whether ProTTA effectively encodes MD information that benefits function prediction beyond what sequence alone provides.

We also include models proposed for embedding MD trajectories in tasks other than function prediction: ProDAR (Chiang et al., 2022), ESTAG (Wu et al., 2023), and EGNO (Xu et al., 2024). Although not designed for function prediction, they receive the same task-specific adaptations as ProTTA (detailed in Appendix C.2), which isolates whether the temporal triangular attention (TTA) design of ProTTA is superior to prior MD embedding methods.

Dynamics Forecasting Baselines We include two baselines designed for MD forecasting: (1) MDGen (Jing et al., 2024b), a flow-based generative model that captures cross-frame relationships; and (2) ConfRover (Shen et al., 2025), a causal late-fusion model built on the sequential and MD information.

4.3 EXPERIMENTAL SETTINGS

For function prediction, we report F1, AUPRC, and AUROC on the aggregated HPA labels, and $F_{\max}$, $S_{\min}$, and term-centric AUPR on the GO labels, following the CAFA challenges (Zhou et al., 2019). For dynamics forecasting, following ConfRover (Shen et al., 2025), we report the Jensen–Shannon divergence (JSD) between the predicted and ground-truth coordinate distributions, recall and F1 under distance thresholds, and RMSD.

For function prediction, we combine every MD embedding model, including ProTTA, with ESM-2 sequence embeddings through late fusion, since sequence information is complementary to MD for function prediction. For dynamics forecasting, we do not evaluate the other MD embedding models on this task, because adapting them to generate per-residue coordinates, unlike predicting protein-level labels, is non-trivial. We set the TTA window to 8 frames and report results with other window lengths in Section 4.5.

4.4 RESULTS

Function Prediction As shown in Tables 1, 2, and 3, ProTTA achieves the best performance on 22 of the 24 metrics across both benchmarks, outperforming both sequence-based models and MD

Table 1: GO term classification results on mdCATH. Best results are in **bold**; second-best results are underlined.

	MF			BP			CC		
	Fmax	Smin	AUPR	Fmax	Smin	AUPR	Fmax	Smin	AUPR
Sequence-based Models									
ESM-2	.423	<u>13.9</u>	.269	.423	44.5	.282	.562	17.0	.382
SaProt	.368	15.0	.185	.378	48.3	.212	.521	18.6	.301
SeqDance	.368	15.2	.177	.354	49.7	.180	.514	19.3	.274
MD Embedding Models, fused with ESM-2									
ProDAR	.424	14.2	.253	.418	45.2	<u>.289</u>	.569	17.0	<u>.397</u>
ESTAG	.410	14.2	.262	<u>.427</u>	<u>44.3</u>	.284	.570	16.9	.391
EGNO	<u>.429</u>	<u>13.9</u>	<u>.276</u>	.416	44.4	.279	<u>.571</u>	<u>16.7</u>	.395
ProTTA (ours)	**.482**	**13.3**	**.333**	**.454**	**42.5**	**.311**	**.610**	**15.7**	**.440**

Table 2: GO term classification results on ATLAS. Best results are in **bold**; second-best results are underlined.

	MF			BP			CC		
	Fmax	Smin	AUPR	Fmax	Smin	AUPR	Fmax	Smin	AUPR
Sequence-based Models									
ESM-2	.575	5.74	.405	.492	25.0	.370	.646	11.6	.456
SaProt	.583	5.94	.368	**.525**	25.7	.396	.648	11.0	.455
SeqDance	.555	6.62	.327	.451	30.1	.269	.603	13.6	.372
MD Embedding Models, fused with ESM-2									
ProDAR	.561	5.94	.401	.483	25.7	.370	.641	11.7	.461
ESTAG	.571	6.52	.295	.420	29.4	.242	.564	13.1	.347
EGNO	.553	6.72	.302	.418	29.1	.258	.560	13.2	.367
ProTTA (ours)	**.597**	**5.56**	**.418**	.496	**24.2**	**.409**	**.650**	**10.8**	**.465**

Table 3: HPA classification results. Best results are in **bold**; second-best results are underlined.

	mdCATH			ATLAS		
	F1	AUPRC	AUROC	F1	AUPRC	AUROC
Sequence-based Models						
ESM-2	.683	.740	.870	.490	.507	.747
SaProt	.555	.606	.778	.478	.515	**.774**
SeqDance	.550	.577	.772	.440	.450	.702
MD Embedding Models, fused with ESM-2						
ProDAR	.677	.739	.870	.493	.515	.742
ESTAG	.693	.743	.872	.487	.509	.742
EGNO	.676	.752	.878	.499	.510	.740
ProTTA (ours)	**.707**	**.771**	**.886**	**.513**	**.517**	.750

embedding models. The gain is more pronounced on mdCATH, which contains more proteins and function labels. Compared with ESM-2 alone, ProTTA improves HPA F1 by 0.024 (3.5% relative) and GO $F_{\max}$ by 0.059 on MF (13.9%), 0.031 on BP (7.3%), and 0.048 on CC (8.5%). The gains on ATLAS are also consistent, with relative improvements of 4.7% in HPA F1 and 3.8%, 0.8%, and 0.6% in GO $F_{\max}$ on MF, BP, and CC, respectively. MF benefits the most from ProTTA, which is consistent with the fact that MF terms largely describe the physical behavior of the protein itself, such as catalytic activity, ion binding, and transporter activity. Notably, whereas most MD embedding models yield only marginal gains over ESM-2 or even degrade its performance, ProTTA improves over ESM-2 on every metric of both benchmarks, confirming that ProTTA extracts information from MD trajectories that benefits function prediction.

Dynamics Forecasting Table 4 reports the 100 ns forecasting results. We quote ConfRover performances from their paper and reproduced MDGen performances. ProTTA outperforms MDGen, achieving a 4.7% lower JSD, 36.2% higher recall, 24.2% higher F1, and 20.2% lower RMSD in relative terms. Moreover, ProTTA performs close to ConfRover, which additionally incorporates sequence information. We further perform time-lagged independent component analysis (tICA) and compare ProTTA against MDGen and ground-truth MD trajectories (Figure 4). ProTTA outperforms MDGen at all lag times, with the largest margin at lags within 8 frames. This matches the 8-frame TTA window, confirming that ProTTA effectively captures cross-frame information.

Table 4: 100 ns forecasting results on ATLAS. ProTTA outperforms MDGen and performs comparably to ConfRover, which additionally uses sequence input.

	JSD	Recall	F1	RMSD
MD-only				
MDGen	.569	.293	.434	1.09
ProTTA	.542	.399	.539	0.87
MD + Sequence				
ConfRover	.51	.42	.58	-
Reference				
MD replica	.304	.684	.79	.00

Figure 4: tICA correlation at different lag frames for ProTTA, MDGen, and a ground-truth MD replica. ProTTA shows the largest improvement within 8 frames, matching its 8-frame TTA window.

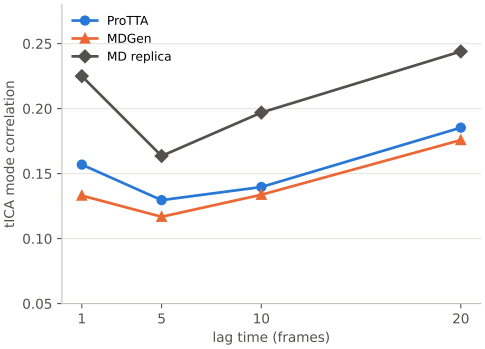

Table 5: Ablation results on HPA classification (F1) and per-category GO classification (term-centric AUPR). *Non-temporal*: TTA replaced by per-frame triangular attention followed by late fusion; *no pretraining*: trained from scratch; *4-/2-frame window*: shortened TTA window.

	mdCATH				ATLAS			
	HPA	MF	BP	CC	HPA	MF	BP	CC
ProTTA	.707	.333	.311	.440	.513	.418	.409	.465
4-frame window	.683	.316	.306	.422	.504	.398	.380	.458
2-frame window	.674	.325	.304	.429	.494	.389	.379	.450
non-temporal	.678	.302	.307	.423	.490	.392	.371	.460
no pretraining	.700	.323	.303	.426	.508	.412	.391	.462

4.5 Model Ablations

To isolate the contributions of TTA and our pretraining initialization from the other modules of ProTTA, we perform the following ablations: (1) replacing TTA with vanilla per-frame triangular attention followed by late fusion across frames; (2) keeping the architecture of ProTTA but training it from scratch; and (3) shortening the TTA window to 4 and 2 frames. Results are shown in Table 5. Shortening the TTA window consistently degrades performance, as does removing the temporal component entirely. Interestingly, in several cases a 2-frame window hurts performance even more than removing temporal modeling altogether. One possible explanation is that, because MD trajectories are sampled from a physical process, short windows are dominated by thermal fluctuations rather than meaningful conformational motion. Removing pretraining also degrades performance, although generally less than removing the temporal component. This supports our hypothesis that pretraining on molecular dynamics also benefits protein-level function prediction.

5 Conclusions

This paper addresses an open problem in protein analysis: information from molecular dynamics (MD) remains underutilized, particularly for function prediction. We identify several challenges in leveraging this information that arise from the inherent complexity of MD data, and propose to address them by capturing motion information across frame-wise embeddings. To this end, we introduce temporal triangular attention (TTA), a novel mechanism that updates pairwise structure embeddings with traces of residue movement. Built on TTA, ProTTA achieves state-of-the-art performance on function classification and improves MD-only trajectory simulation. Nonetheless, due to its computational cost, ProTTA is currently limited to capturing short-timescale MD dynamics; extending it to long-timescale MD and to other protein analysis tasks remains an open direction.

6 AI USE STATEMENT

During this study, we use AI for: implement methods, clean and reformat dataset, edit the paper to improve readability, formulate mathematical claims, assist in the writing of proofs, create or modify scientific figures or images. We haven't used AI for: design or provide feedback on research methodology or experiments, draft parts of a research paper, interpret results, propose or refine hypotheses. And these tasks do not apply to this paper: Generate synthetic data sets, help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, assist with translation, support qualitative and thematic data analysis,

7 REPRODUCIBILITY STATEMENT

We use public datasets with detailed dataset establishment and model specifications in Appendix. We further provide our codes in `https://anonymous.4open.science/r/ProTTA-149D`

REFERENCES

Uniprot: the universal protein knowledgebase in 2025. *Nucleic acids research*, 53(D1):D609–D617, 2025.

The gene ontology knowledgebase in 2026. *Nucleic acids research*, 54(D1):D1779–D1792, 2026.

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.

Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, et al. Openfold: retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature methods*, 21(8):1514–1524, 2024.

David D Boehr, Ruth Nussinov, and Peter E Wright. The role of dynamic conformational ensembles in biomolecular recognition. *Nature chemical biology*, 5(11):789–796, 2009.

Yuan Chiang, Wei-Han Hui, and Shu-Wei Chang. Encoding protein dynamic information in graph representation for functional residue identification. *Cell Reports Physical Science*, 3(7), 2022.

Elan Z Eisenmesser, Oscar Millet, Wladimir Labeikovsky, Dmitry M Korzhnev, Magnus Wolf-Watz, Daryl A Bosco, Jack J Skalicky, Lewis E Kay, and Dorothee Kern. Intrinsic dynamics of an enzyme underlies catalysis. *Nature*, 438(7064):117–121, 2005.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Chao Hou, Haiqing Zhao, and Yufeng Shen. Protein language models trained on biophysical dynamics inform mutation effects. *Proceedings of the National Academy of Sciences*, 123(4): e2530466123, 2026. doi: 10.1073/pnas.2530466123. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2530466123`.

PH Hünenberger, AE Mark, and WF Van Gunsteren. Fluctuation and cross-correlation analysis of protein motions observed in nanosecond molecular dynamics simulations. *Journal of molecular biology*, 252(4):492–503, 1995.

Toshiko Ichiye and Martin Karplus. Collective motions in proteins: a covariance analysis of atomic fluctuations in molecular dynamics and normal mode simulations. *Proteins: Structure, Function, and Bioinformatics*, 11(3):205–217, 1991.

Yaan J Jang, Qi-Qi Qin, Si-Yu Huang, Arun T John Peter, Xue-Ming Ding, and Benoît Kornmann. Accurate prediction of protein function using statistics-informed graph networks. *Nature communications*, 15(1):6601, 2024.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=rs8Sh2UASt.

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. *Advances in Neural Information Processing Systems*, 37:40534–40564, 2024b.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Dan Kalifa, Eric Horvitz, and Kira Radinsky. Learning protein representations with conformational dynamics. *Bioinformatics*, 42(5):btag254, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Maxat Kulmanov, Francisco J Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T Arold, and Robert Hoehndorf. Protein function prediction as approximate semantic entailment. *Nature Machine Intelligence*, 6(2):220–228, 2024.

Boqiao Lai and Jinbo Xu. Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, 23(1):bbab502, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Antonio Mirarchi, Toni Giorgino, and Gianni De Fabritiis. mdcath: A large-scale md dataset for data-driven computational biophysics. *Scientific Data*, 11(1):1299, 2024.

Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227, 2013.

Yuning Shen, Lihao Wang, Huizhuo Yuan, Yan Wang, Bangji Yang, and Quanquan Gu. Confrover: Simultaneous modeling of protein conformation and dynamics via autoregression. *arXiv preprint arXiv:2505.17478*, 2025.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations*, volume 2024, pp. 6987–7009, 2024.

Mathias Uhlén, Linn Fagerberg, Björn M Hallström, Cecilia Lindskog, Per Oksvold, Adil Mardinoglu, Åsa Sivertsson, Caroline Kampf, Evelina Sjöstedt, Anna Asplund, et al. Tissue-based map of the human proteome. *Science*, 347(6220):1260419, 2015.

Yann Vander Meersche, Gabriel Cretin, Aria Gheeraert, Jean-Christophe Gelly, and Tatiana Galochkina. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic acids research*, 52(D1):D384–D392, 2024.

Wenkang Wang, Yunyan Shuai, Min Zeng, Wei Fan, and Min Li. Dpfunc: accurately predicting protein function via deep learning with domain-guided structure information. *Nature communications*, 16(1):70, 2025.

Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. *Advances in Neural Information Processing Systems*, 36:45360–45380, 2023.

Minkai Xu, Jiaqi Han, Aaron Lou, Jean Kossaifi, Arvind Ramanathan, Kamyar Azizzadenesheli, Jure Leskovec, Stefano Ermon, and Anima Anandkumar. Equivariant graph neural operator for modeling 3d dynamics. *arXiv preprint arXiv:2401.11037*, 2024.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):244, 2019.

A TRIANGULAR MECHANISM

Triangular self-attention. This operator updates $\mathbf{z}_{ij}$ by attending over all third residues k, following the intuition that the relation between i and j should be informed by every pair of edges (i,k) and (j,k). For attention head h of width c, let $\mathbf{q}_{ij}^h, \mathbf{k}_{ij}^h, \mathbf{v}_{ij}^h = \mathrm{Linear}(\mathbf{z}_{ij}) \in \mathbb{R}^c$ be the query, key and value, $b_{ij}^h = \mathrm{Linear}(\mathbf{z}_{ij}) \in \mathbb{R}$ a scalar pair bias, and $\mathbf{g}_{ij}^h = \sigma(\mathrm{Linear}(\mathbf{z}_{ij})) \in \mathbb{R}^c$ a gate. The attention-based update is calculated via:

$$\alpha_{ijk}^h = \mathrm{softmax}_k \left(\tfrac{1}{\sqrt{c}} \, \mathbf{q}_{ij}^{h\,\top} \mathbf{k}_{ik}^h + b_{jk}^h \right), \tag{9}$$

$$\tilde{\mathbf{z}}_{ij} = \mathbf{W}^o \, \mathrm{concat}_h \Big(\mathbf{g}_{ij}^h \odot \sum_{k=1}^{L} \alpha_{ijk}^h \, \mathbf{v}_{ik}^h \Big). \tag{10}$$

Edge ij thus attends to the edges ik that share its starting node, and the bias b_{jk}^h from the third edge closes the triangle. The update $\tilde{\mathbf{z}}_{ij}$ is added to $\mathbf{z}_{ij}$ residually. A symmetric variant around the ending node attends to the edges kj with bias b_{ki}^h.

Triangular multiplicative update. This operator encourages geometric consistency within each triangle by multiplying gated projections of its two other edges. With $\mathbf{a}_{ij}, \mathbf{b}_{ij} = \sigma(\mathrm{Linear}(\mathbf{z}_{ij})) \odot \mathrm{Linear}(\mathbf{z}_{ij})$ and a gate $\mathbf{g}_{ij} = \sigma(\mathrm{Linear}(\mathbf{z}_{ij})) \in \mathbb{R}^{c_z}$ with its own parameters, the outgoing and incoming updates aggregate over the third residue k:

$$\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathbf{W}^o \, \mathrm{LN} \Big(\sum_{k=1}^{L} \mathbf{a}_{ik} \odot \mathbf{b}_{jk} \Big) \text{ (outgoing)}, \tag{11}$$

$$\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathbf{W}^o \, \mathrm{LN} \Big(\sum_{k=1}^{L} \mathbf{a}_{ki} \odot \mathbf{b}_{kj} \Big) \text{ (incoming)}. \tag{12}$$

B GEOMETRIC MOTIVATION AND ARCHITECTURAL DISTINCTION

The main contribution of TTA is architectural. We use the elementary observation below to motivate cross-frame candidates and extending triangle attention to temporal dimension.

After superposing all frames of a trajectory window onto a common reference, let $\mathbf{x}_i(t) \in \mathbb{R}^3$ denote the position of residue i at frame t. Let $d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$ be the the target-edge length and the following is

$$d_{ik}(t,u) = \|\mathbf{x}_i(t) - \mathbf{x}_k(u)\|. \tag{13}$$

cross-frame edge length. Since $\mathbf{x}_i(t)$, $\mathbf{x}_j(t)$, and $\mathbf{x}_k(u)$ are points in the same aligned coordinate system, the ordinary triangle inequality gives

$$d_{ij}(t) \le d_{ik}(t,u) + d_{jk}(t,u). \tag{14}$$

Proposition 1 states this observation formally and gives the proof.

Thus the position of residue k at any frame u is a valid geometric reference point for the directed target edge $i \to j$ at frame t. For $u = t$, this is the familiar within-frame triangle used to motivate static triangular attention; allowing $u \neq t$ adds cross-frame candidates. This observation motivates exposing (k,u) jointly to the operator, but it does not prescribe how candidates should be scored. In particular, TTA's learned compatibility scores are not constrained to equal the distances in equation 13.

We further provide propositions to formalize the claims of elementary geometric fact.

Proposition 1 (Cross-frame triangle inequality) *Suppose all trajectory frames have been superposed onto a common coordinate system. For any residues i, j, k and frames t, u, define*

$$d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|, \qquad d_{ik}(t,u) = \|\mathbf{x}_i(t) - \mathbf{x}_k(u)\|.$$

Then

$$d_{ij}(t) \leq d_{ik}(t,u) + d_{jk}(t,u).$$

Proof. The three coordinates $\mathbf{x}_i(t)$, $\mathbf{x}_j(t)$, and $\mathbf{x}_k(u)$ are points in $\mathbb{R}^3$. The upper bound is the Euclidean triangle inequality. □

This proposition shows only that a cross-frame position is a geometrically valid third vertex after superposition.

C Implementation Details

C.1 Dataset

For data splitting, we follow the setting of AlphaFlow (Jing et al., 2024a) to split trajectory forecasting data, maintaining the same as both baselines. For function prediction, we perform a 60:20:20 split while ensuring GO label balance across the subsets. We pretrain ProTTA using all proteins in either dataset excluding those used for testing set of the two tasks.

We gather protein function labels from UniProt database (uni, 2025), with further formulating into two schemes.

4-HPA We refer to human protein atlas database [1] and aggregate four categories: Enzymes, metal binding, membrane proteins, and transporters. The sub-categories corresponding to each category is shown in Table 6.

Table 6: HPA categories and corresponding sub-category terms.

Category	Sub-terms
Enzyme	hydrolase, transferase, kinase, ligase, lyase, isomerase, oxidoreductase, protease, peptidase, nuclease
Metal Binding	metal-binding, zinc, iron, calcium, copper, manganese, heme
Membrane Protein	membrane, transmembrane, cell membrane, GPI-anchor
Transporters	transport, transporter, symport, antiport

GO We refer to gene ontology (GO) annotations to build the labels for each protein. Moreover, we cutoff terms with fewer than 50 positive cases in the training split, following CAFA challenge (Zhou et al., 2019).

C.2 ProTTA

We use the following hyperparamers shared across all experiments: 4 blocks of TTA; pair representation channels 64, node channels 64, attention 4 heads × 16, dropout 0.1, and with gradient checkpointing. In each block, triangle multiplication uses hidden size 32, and pair transition is achieved by a MLP 64→128→64, all residual with layer norm. During the calculation of TTA, we limit the (i,k) to be within the closest 16 residues of i. The pair representation is initialised from the C-alpha distance matrix of the window-mean structure and mapped linearly to channels.

For function prediction, we first project pairwise representation to per-residue representation and concatenate with sequence embeddings. The output embedding is then mean-pooled across residues before feeding into a MLP, following approaches of ESM-2 (Lin et al., 2023). We set MLP hidden size to be 512 and 256, with output dimension corresponding to number of function labels. We train the model by binary cross-entropy. We use Adam optimizer, with learning rate $3 * 10^{-4}$ and a weight decay of 10^{-4}.

For trajectory generation, we use a diffusion head as an EDM denoiser $D_\theta(y_\sigma, \sigma \mid s(t), z(t))$ (Karras et al., 2022), which is a 6-layer pre-LayerNorm Transformer over residues with width 256, 8

[1] proteinatlas.org

Table 7: Dataset statistics.

	mdCATH	ATLAS
Proteins	4276	1826
Train / val / test	2992 / 642 / 642	1278 / 274 / 274
Residues, mean (median)	139 (115)	229 (176)
Replicas	5	3
Frames used per replica	64	64
Enzymes	2123 (49.6%)	753 (41.2%)
Metal binding	1364 (31.9%)	557 (30.5%)
Membrane proteins	906 (21.2%)	288 (15.8%)
Transporters	405 (9.5%)	138 (7.6%)
GO terms (MF / BP / CC)	58 / 173 / 59	12 / 56 / 23
GO terms per annotated protein	29.3	18.5
Median positives per GO term	142	115

heads and a $4\times$ feed-forward expansion. Its input token for residue i is the scaled noisy displacement $c_{\text{in}}(\sigma)\, y_{\sigma,i}$ concatenated with $s_i(t)$ and linearly projected, plus an embedding of the noise level (Fourier features of $\ln \sigma/4$). Every layer adds a per-head attention bias obtained by a linear projection of $\text{LN}(z_{ij}(t))$, so the pair representation steers how residues attend to each other. We use the EDM preconditioning $D_\theta = c_{\text{skip}}(\sigma)\, y_\sigma + c_{\text{out}}(\sigma)\, F_\theta(\cdot)$ with $\sigma_{\text{data}} = 1.25$ Å, and the output layer is zero-initialised. We train the forecaster with Adam (learning rate 3×10^{-4}, weight decay 0.01), a linear warm-up over the first 1000 steps followed by a constant learning rate, gradient clipping at norm 1.0, and bf16.

C.3 Pretraining

We pretrain the model to predict dynamic cross-correlation matrix over the next window. The coupling head is a linear map on pairwise representation. Loss is 1 - pearson r over valid off-diagonal pairs, averaged over the batch. We use AdamW, lr $3 * 10^{-4}$, weight decay 10^{-4}, one-cycle schedule with 10 % warm-up, gradient clipping 1.0, bf16, batch 8, 16 epochs, checkpoint selected on validation Pearson agreement.

C.4 Baselines

We adapt the original embeddings of sequence-based models and apply the readout following ESM-2. For MD embedding models, EGNO and ESTAG are run from the official code on C-alpha-only residue graphs, pretrained with their native forecasting objective on the same window as ours. ProDAR and SDF are instead trained end-to-end with the classification head. All models receive the per-residue downstream classification head as ProTTA.

C.5 Hardware

A majority of experiments are conducted on NVIDIA A100 40 GB with 32–64 GB RAM. Some of the experiments are conducted on A40 48 GB and H200 GPUs.