Triangle Multiplication is All You Need for Biomolecular Structure Representations

Jeffrey Ouyang-Zhang[1,2] *, **Pranav Murugan**[1], **Daniel J. Diaz**[2], **Gianluca Scarpellini**[1], **Richard Strong Bowen**[1], **Nate Gruver**[1], **Adam Klivans**[2], **Philipp Krähenbühl**[2], **Aleksandra Faust**[1], **Maruan Al-Shedivat**[1]
[1]Genesis Research [2]UT Austin

Abstract

AlphaFold has transformed protein structure prediction, but emerging applications such as virtual ligand screening, proteome-wide folding, and de novo binder design demand predictions at a massive scale, where runtime and memory costs become prohibitive. A major bottleneck lies in the Pairformer backbone of AlphaFold3-style models, which relies on computationally expensive triangular primitives—especially triangle attention—for pairwise reasoning. We introduce *Pairmixer*, a streamlined alternative that eliminates triangle attention while preserving higher-order geometric reasoning capabilities that are critical for structure prediction. *Pairmixer* substantially improves computational efficiency, matching state-of-the-art structure predictors across folding and docking benchmarks, delivering up to 4× faster inference on long sequences while reducing training cost by 34%. Its efficiency alleviates the computational burden of downstream applications such as modeling large protein complexes, high-throughput ligand and binder screening, and hallucination-based design. Within BoltzDesign, for example, *Pairmixer* delivers over 2× faster sampling and scales to sequences ∼30% longer than the memory limits of Pairformer.

1 Introduction

AlphaFold (Senior et al., 2020; Jumper et al., 2021) has transformed protein structure prediction and become an indispensable tool across the biological sciences. Yet emerging applications increasingly demand massive scale. Virtual screening of protein–ligand interactions, modeling of large protein complexes, proteome-wide folding, and iterative de novo binder design already require millions (and soon billions) of inference calls. At this scale, runtime and memory efficiency are critical bottlenecks: for example, Boltz-1 (Wohlwend et al., 2024) requires over 15 minutes to process a single 2048-token sequence on an A100 GPU (see Section 5.3). The dominant computational cost comes from pairwise token representations and triangular primitives, which scale *cubically* with sequence length L. While triangle multiplication is implemented efficiently via matrix multiplications, triangle attention requires L attention operations, introducing substantial memory and runtime overhead.

We introduce *Pairmixer*, a streamlined alternative to the Pairformer backbone of AlphaFold3 (Abramson et al., 2024). By retaining triangle multiplication and feed-forward networks while eliminating triangle and sequence attentions, *Pairmixer* preserves the ability to reason over higher-order geometric interactions that are critical for structure prediction while alleviating Pairformer's heavy computational burden. Despite this simplification, *Pairmixer* performs comparably on RCSB and CASP15 test sets against state-of-the-art predictors such as AlphaFold, Chai-1, and Boltz-1, while providing 4× faster inference on long sequences. It matches the performance of Pairformer backbone across protein folding, protein–protein docking, and protein–ligand docking, while training in 34% fewer GPU-days across multiple model sizes (see Figure 1).

By reducing both runtime and memory requirements, *Pairmixer* expands the scope of feasible downstream applications of structure prediction. It enables modeling of larger protein complexes

*Work done during an internship at Genesis Research

{jozhang,danny.diaz,klivans,philkr}@cs.utexas.edu
{pranav,gianscarpe,richard,ngruver,sandra,maruan}@genesistherapeutics.ai

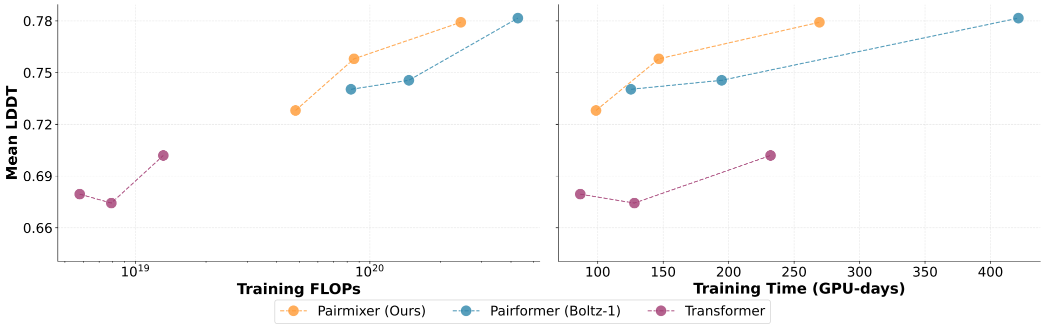


Figure 1: ***Pairmixer* is an efficient architecture for biomolecular structure prediction.** Across multiple model sizes, *Pairmixer* matches the performance of the leading Pairformer architecture while delivering greater training efficiency.

beyond the limits of triangle attention, supports high-throughput screening of ligands and binders, and accelerates hallucination-based design pipelines (Pacesa et al., 2025). Within the BoltzDesign1 (Cho et al., 2025) framework, *Pairmixer* provides over 2× faster sampling and scales to sequences beyond 800 amino acids, where BoltzDesign otherwise fails due to memory overflow. Our analysis highlights the role of the pair representation in learning precise distances between residues and suggests that triangle multiplication learns to capture sparse interactions among residue triplets.

2 Related Work

Biomolecular Structure Prediction. Protein structure prediction has progressed rapidly in recent years, with early efforts primarily focused on modeling monomeric proteins (Senior et al., 2020; Jumper et al., 2021; Baek et al., 2021; Yang et al., 2020; Ahdritz et al., 2024). As these approaches matured, structure predictors expanded to handle multimeric assemblies (Evans et al., 2021; Baek et al., 2023) and other modalities such as nucleic acids (Baek et al., 2024). Today, state-of-the-art predictors can fold complexes that span a wide range of biomolecular types (Abramson et al., 2024; IntFold et al., 2025; Boitreaud et al., 2024; Wohlwend et al., 2024; ByteDance et al., 2025).

Biomolecular structure predictors rely on specialized backbones that capture complex geometric relationships among molecular entities. Early approaches such as trRosetta (Yang et al., 2020) and AlphaFold1 (Senior et al., 2020) leveraged convolutional neural networks to extract pairwise residue features from multiple sequence alignments (MSAs) and predict inter-residue distances. AlphaFold2 introduced the transformer-based Evoformer to jointly model MSA and pair representations (Jumper et al., 2021), while AlphaFold3 refined it with the Pairformer, which decouples MSA and pair processing (Abramson et al., 2024). The Pairformer has since become the de-facto backbone architecture for biomolecular structure prediction (IntFold et al., 2025; Boitreaud et al., 2024; Wohlwend et al., 2024; ByteDance et al., 2025). However, despite its strong performance, the Pairformer remains complex and computationally demanding.

Several alternative architectures have been proposed to simplify structure prediction backbones. MiniFold (Wohlwend et al., 2025) streamlines Alphafold2's Evoformer using a lightweight Miniformer based on triangle multiplications. SimpleFold (Wang et al., 2025) replaces the Evoformer with a sequence-only transformer that omits pair representations. Our work also simplifies backbone design, but unlike prior efforts focused on monomeric folding, *Pairmixer* is developed for AlphaFold3-like cofolding models, enabling structure prediction across broader biomolecular modalities.

Downstream Applications of Structure Prediction. The success of biomolecular structure prediction has enabled a growing number of downstream applications, many of which leverage predicted structures at unprecedented scales. Large-scale resources such as the AlphaFold Database (Varadi et al., 2022) and OpenFold (Ahdritz et al., 2024) have generated massive synthetic protein structure datasets using AlphaFold2, powering advances in structure search (Van Kempen et al., 2024), protein language modeling (Heinzinger et al., 2024; Ouyang-Zhang et al., 2024; Hayes et al., 2025), and diffusion-based structure generation (Geffner et al., 2025; Lin et al., 2024; Daras et al., 2025). Structure predictors now drive virtual screening pipelines that evaluate millions of candidate drugs based on predicted protein–ligand interactions (Wong et al., 2022; Shamir & London, 2025; Scardino

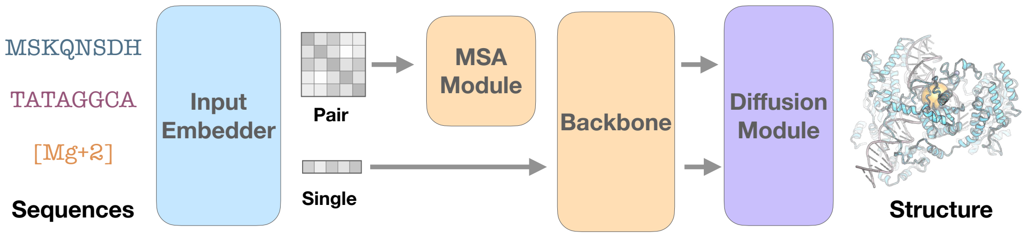


Figure 2: **Overview of Biomolecular Structure Prediction.** Given a list of sequences, our model predicts the 3D folded structure of all sequences within a single complex. Input sequences are first *embedded* into a single representation for each residue and a pair representation to capture the relationship between pairs of residues. The *MSA Module* and *Backbone* (e.g., Pairformer) extracts deep pairwise features capturing inter-residue interactions, which are then passed to the *diffusion module* to generate the 3D structure. (Additional inputs such as MSAs, conformers, and templates are omitted for clarity.)

et al., 2023), and large-scale folding studies that map the human interactome (Ille et al., 2025; Zhang et al., 2025). Hallucination-based generation methods such as BindCraft (Pacesa et al., 2025) further use predictors in iterative optimization loops requiring millions of model evaluations. As these applications expand in scope and scale, inference speed becomes a critical bottleneck. We introduce a structure predictor that matches state-of-the-art accuracy while operating at a fraction of the runtime, enabling faster and broader deployment of downstream workflows.

Attention-free Architectures. While transformers lead modern architectures, attention-free variants aim to improve scalability. FNet (Lee-Thorp et al., 2021) and related models (Poli et al., 2023; Zhai et al., 2021) replace attention with Fourier or convolutional mixing for sub-quadratic efficiency, while MLP-Mixer (Tolstikhin et al., 2021) achieves competitive performance using token- and channel-wise multi-layer perceptrons (MLPs). *Pairmixer* removes attention entirely from the backbone and mixes tokens through matrix multiplication.

Architectures based on triangle multiplication have been explored in several prior works. Genie2 (Lin et al., 2024) performs de-novo structure generation by iteratively updating a pair representation through triangle multiplications, while MSA Pairformer (Akiyama et al., 2025) applies similar operations to extract features from multiple sequence alignments. IgFold (Ruffolo et al., 2023) incorporates triangle operations within GNN layers. *Pairmixer* likewise learns rich protein representations through triangle multiplication, but in the context of biomolecular structure prediction.

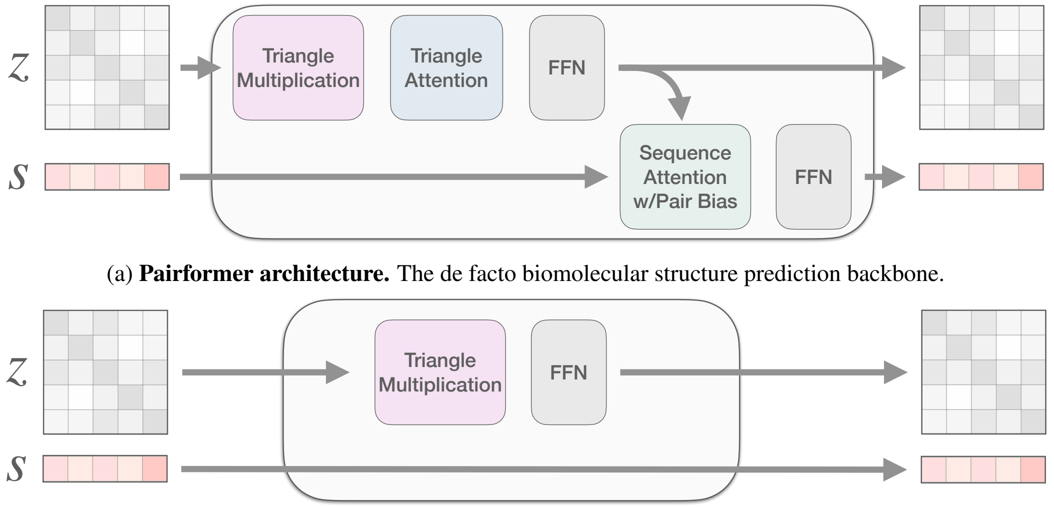
3 Preliminaries

Let $x = \{x^{(1)}, \cdots, x^{(K)}\}$ denote a collection of K biomolecular sequences. Each sequence $x^{(k)} = (x_1^{(k)}, \cdots, x_{L^{(k)}}^{(k)})$ consists of tokens $x_i^{(k)} \in \mathcal{T}$ corresponding to an amino acid, a nucleic acid, or small molecule heavy atoms. $L^{(k)}$ denotes the number of tokens in biomolecule $x^{(k)}$. The goal of biomolecular structure prediction is to map the sequences x to a three-dimensional structure $a = \{a^{(1)}, \cdots, a^{(K)}\}$, where each biomolecular structure $a^{(k)} = (\boldsymbol{a}_1^{(k)}, \cdots, \boldsymbol{a}_{N^{(k)}}^{(k)})$ consists of atomic coordinates $\boldsymbol{a}_j^{(k)} \in \mathbb{R}^3$, and $N^{(k)}$ denotes the number of atoms in biomolecule k. See Figure 2 for an overview.

The Input Embedder concatenates the sequences $x = \{x^{(1)}, \dots, x^{(K)}\}$ and embeds it into a "*single*" length $L = \sum_{k=1}^{K} L^{(k)}$ sequence representation $\boldsymbol{s}^{\text{init}} \in \mathbb{R}^{L \times C_s}$ of dimension C_s. Modern structure predictors (Jumper et al., 2021) additionally initialize a "*pair*" representation $\boldsymbol{z}^{\text{init}} \in \mathbb{R}^{L \times L \times C_z}$:

$$\boldsymbol{z}_{ij} = \boldsymbol{s}_i + \boldsymbol{s}_j + \mathbf{PE}(i, j),$$

where $\mathbf{PE}(i, j)$ is a positional encoding that incorporates both intra- and inter-sequence distances and C_z is the pair embedding dimension. Intuitively, $\boldsymbol{z}_{ij} \in \mathbb{R}^{C_z}$ captures the relational context between tokens $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ and enables the model to reason about longer-range couplings. Since pairwise reasoning is critical for structure prediction, we adopt the same input embedding scheme.



(a) **Pairformer architecture.** The de facto biomolecular structure prediction backbone.

(b) ***Pairmixer* architecture.** An efficient yet effective biomolecular structure prediction backbone.

Figure 3: **Schematic comparison of the Pairformer and *Pairmixer* backbones.** *Pairmixer* simplifies the Pairformer architecture by removing redundancies. This results in faster training and inference, expanding the scale of downstream applications.

The MSA Module encodes evolutionary information that is crucial for structure prediction (Benner & Gerloff, 1991; Yanofsky et al., 1964; Ovchinnikov et al., 2017; 2014; Morcos et al., 2011; Weigt et al., 2009). For each amino acid or nucleic acid sequence $x^{(k)}$, we perform a homology search to construct a multiple sequence alignment (MSA) of related sequences that likely adopt the same fold. Formally, $\mathbf{MSA}(x^{(k)}) \in (\mathcal{T} \cup \{\texttt{GAP}\})^{M^{(K)} \times L^{(K)}}$ contains $M^{(k)}$ aligned sequences of length $L^{(k)}$. This alignment establishes positional correspondence across homologous sequences, enabling detection of conserved sites and co-evolutionary couplings. The resulting MSAs are then paired, concatenated, and embedded into $\boldsymbol{m}^{\text{init}} \in \mathbb{R}^{M \times L \times C_m}$ where M is the number of filtered homologous sequences and C_m is the MSA embedding dimension.

The MSA module takes $(\boldsymbol{m}^{\text{init}}, \boldsymbol{z}^{\text{init}})$ as input, extracts structurally-relevant evolutionary patterns from $\boldsymbol{m}^{\text{init}}$, and encodes pairwise interactions into $\boldsymbol{z}^{\text{msa}}$ to guide folding. Since processing all M sequences in the MSA is computationally expensive, AlphaFold3 introduced a shallow 4-layer MSA module after which the MSA is discarded while the evolutionary-aware pair representation $\boldsymbol{z}^{\text{msa}}$ continues to be refined. Our model derives $\boldsymbol{z}^{\text{msa}}$ from an MSA module but introduces a more efficient feature extractor to refine its evolutionary signals.

The Pairformer backbone serves as the primary feature extractor for AlphaFold3 (Abramson et al., 2024), producing structrually-aware representations that encode geometric constraints between residues (see Figure 3a). It takes $(\boldsymbol{s}^{\text{init}}, \boldsymbol{z}^{\text{msa}})$ as input and employs several specialized modules that iteratively update the sequence and pair representations to produce $(\boldsymbol{s}^{\text{backbone}}, \boldsymbol{z}^{\text{backbone}})$. See Figure 10 for a more detailed treatment of the entire architecture.

The Pairformer contains two specialized modules for processing the pair representation: triangle attention and triangle multiplication. These modules treat the pair representation $\boldsymbol{z} \in \mathbb{R}^{L \times L \times C_z}$ as edge features of a fully-connected graph of L nodes and reason over triplets of residues (nodes) to learn geometric constraints.

Triangle attention computes attention (with pair bias) along every row (and column) of the pair representation. Formally, the update to row i is

$$\mathbf{TriAtt}(\boldsymbol{z})_i = \text{softmax}\Big((\boldsymbol{W}_Q \boldsymbol{z}_i)(\boldsymbol{W}_K \boldsymbol{z}_i)^\top + \boldsymbol{W}_B \boldsymbol{z}\Big)\, \boldsymbol{W}_V \boldsymbol{z}_i$$

Algorithm 1 *Pairmixer* Backbone

```
Require: Input pair representation $\boldsymbol{z}^{\text{msa}} \in \mathbb{R}^{L \times L \times C_z}$
Require: Number of backbone layers $N$
Ensure: Updated pair representation $\boldsymbol{z}_N$
$\boldsymbol{z}_0 \leftarrow \boldsymbol{z}^{\text{msa}}$
for $l = 0$ to $N - 1$ do
    $\boldsymbol{z}_l \leftarrow \boldsymbol{z}_l + \mathbf{TriMulIncoming}(\boldsymbol{z}_l)$
    $\boldsymbol{z}_l \leftarrow \boldsymbol{z}_l + \mathbf{TriMulOutgoing}(\boldsymbol{z}_l)$
    $\boldsymbol{z}_{l+1} \leftarrow \boldsymbol{z}_l + \mathbf{FFN}(\boldsymbol{z}_l)$
end for
return $\boldsymbol{z}_N$
```

where $(\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V)$ are standard attention projection matrices, and $\boldsymbol{W}_B$ projects the pair representation into an attention bias term [1]. $\mathbf{TriAtt}(\boldsymbol{z})_i$ effectively performs attention over all residues while conditioning on residue i.

Triangle multiplication performs matrix multiplications to integrate features across different rows (and columns) of the pair representation. Formally, the update to edge $\boldsymbol{z}_{ij}$ is

$$\mathbf{TriMul}(\boldsymbol{z})_{ij} = \sum_{k=1}^{L} (\boldsymbol{W}_a \boldsymbol{z}_{ik}) \odot (\boldsymbol{W}_b \boldsymbol{z}_{jk})$$

where $\boldsymbol{W}_a, \boldsymbol{W}_b$ are linear projection layers. For each edge $\boldsymbol{z}_{ij}$, triangle multiplication computes how every node k interacts with query nodes i and j through edges $\boldsymbol{z}_{ik}$ and $\boldsymbol{z}_{jk}$.

Both operations scale cubically with sequence length, making the processing of long sequences computationally expensive. Triangle multiplication is more efficient, as it can be implemented with matrix multiplications (e.g., `torch.einsum`), whereas triangle attention incurs the higher cost of L full attention computations. In this work, we streamline the cofolding backbone to its essential components and show that triangle multiplication yields representations as powerful as those from triangle attention, but at substantially lower computational cost, supporting a range of downstream applications.

While the Pairformer is trained with an auxiliary distogram loss that ensures $\boldsymbol{z}^{\text{backbone}}$ accurately represents all pairwise token distances, it does not yet specify an atomic 3-D structure.

The Diffusion Module samples the atomic coordinates conditioned on $(\boldsymbol{s}^{\text{backbone}}, \boldsymbol{z}^{\text{backbone}})$. It uses transformers to derive atomic representations from the token-level sequence and pair representations, and subsequently denoises all-atom coordinates based on these representations. We leverage the diffusion module as-is to realizes 3-D structures conditioned on single and pair representations derived from our efficient backbone.

4 Method

We introduce *Pairmixer*, an attention-free feature extractor for biomolecular structure prediction and design (see Figure 3). *Pairmixer* exclusively updates the pair representation $\boldsymbol{z}^{\text{msa}}$, leaving the single-sequence representation $\boldsymbol{s}^{\text{init}}$ unchanged. Through *triangle multiplication*, *Pairmixer* efficiently mixes features within the pair representation, facilitating reasoning over residue triplets and their geometric constraints. Combined with feed-forward networks (FFN) that process all residue pairs, this architecture provides an effective and expressive backbone for biomolecular structure prediction.

The full algorithmic specification of *Pairmixer* is available in Algorithm 1. In developing *Pairmixer*, we identified and removed two unnecessary modules from the Pairformer: sequence updates and triangle attention.

Removing Sequence Updates. In AlphaFold2's Evoformer backbone, sequence updates were essential components that processed the MSA to capture evolutionary features. However, the MSA

[1] single head and removed scaling for brevity

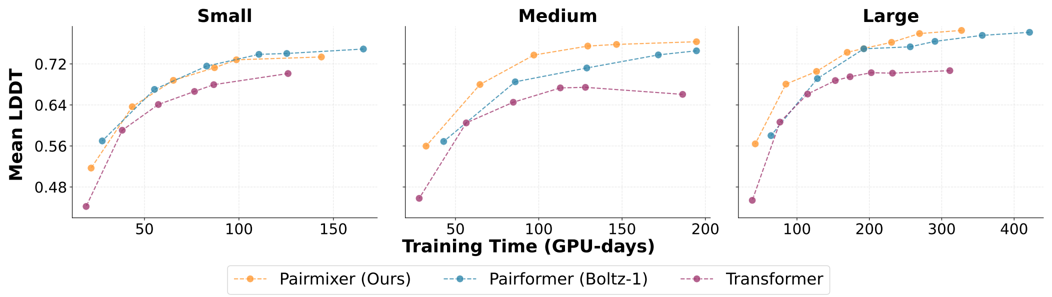


Figure 4: **Performance curves on RCSB test set across model sizes.** We compare three backbone architectures across three model sizes over training. *Pairmixer* matches or surpasses the Pairformer baseline while training more efficiently.

Module in cofolding models now preprocesses the MSA and encodes this evolutionary information directly into the pair representation $\boldsymbol{z}^{\text{msa}}$, eliminating the need for sequence updates to provide evolutionary information. Since the pair updates proved more expressive, we bypass sequence processing entirely and pass the initial sequence representation directly to the diffusion module (i.e., $\boldsymbol{s}^{\text{backbone}} = \boldsymbol{s}^{\text{init}}$).

Removing Triangle Attention. Triangle attention reasons over residue triplets by applying attention to each row of the pair representation $\boldsymbol{z}_i$, using the full $\boldsymbol{z}$ as pairwise bias (see Figure 10b). However, this approach is computationally expensive, requiring L separate attention operations over L tokens per layer. Triangle multiplication offers equivalent capability for capturing geometrically consistent pair representations via a triplet reasoning mechanism, but with significantly lower computational cost. Since both methods have independently demonstrated strong performance in structure prediction (Jumper et al., 2021), we adopt the more efficient triangle multiplication approach.

5 Results

5.1 Implementation Details

We implement *Pairmixer* on top of Boltz-1, an *AlphaFold3* descendant. More specifically, we replace the Pairformer backbone with *Pairmixer* and remove triangle attention from the MSA Module. Note that we do not alter the diffusion module's transformer architecture. We also introduce a transformer baseline that preserves the sequence update while removing the pair update in the backbone. To ensure this baseline is as strong as possible, we modify the architecture to allow features to flow effectively from the MSA module into the diffusion module (see Section A.2).

Following Boltz-1 training schedule (Wohlwend et al., 2024), we train on 384/3456 token/atom crops for the first 53k iterations using the PDB and OpenFold distillation dataset. We then finetune for 15k iterations on the PDB dataset with a larger crop size of 512/4608. To evaluate the generality of our approach, we train models of multiple sizes. Our large configuration matches Boltz-1, with 48 Pairformer layers and 24 diffusion transformer layers. In addition, we develop small and medium variants with 12/24 Pairformer layers and 6/24 diffusion transformer layers, respectively. During inference, we default to 10 recycling steps and 200 sampling steps for all models. In our main evaluation, we sample 5 poses and report the metrics on the top pose (oracle evaluation). Full hyperparameter details are in Table 6.

5.2 Comparisons on Cofolding performance across model sizes

We evaluate our efficient *Pairmixer* architecture against two baselines, Pairformer (Abramson et al., 2024) and a sequence-only Transformer. All models are evaluated on the RCSB test set introduced in Boltz-1 (Wohlwend et al., 2024), which contains 533 structures with at most 40% sequence identity to the training set, maximum small-molecule similarity of 80%, and resolution better than 4.5Å. All models are evaluated at 15, 30, 45, 60, and 68 epochs, totalling 53k iterations and the large model is additionally evaluated during the second phase of 15k iterations. We additionally extend training

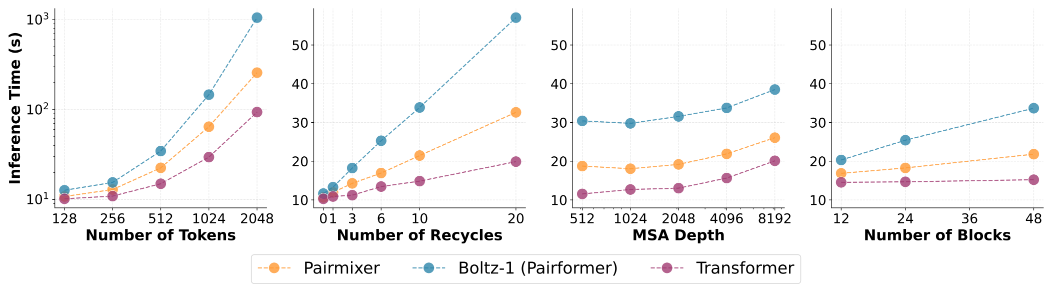


Figure 5: **Inference speed analysis.** We measure runtime across architectures and input sizes. While the Transformer is the fastest overall, *Pairmixer* achieves substantially lower inference times than Pairformer, particularly on longer sequences.

for small and medium *Pairmixer* and Transformer models until the total training time matches the Pairformer. We report the final mean LDDT, averaged across all residues.

Our *Pairmixer* consistently outperforms or matches the Pairformer across all model sizes (see Figure 4). At the large scale, *Pairmixer* reaches Pairformer-level accuracy (mean LDDT of 0.78) while requiring only 66% of the training time. The trend holds at smaller scales: *Pairmixer* surpasses Pairformer at the medium scale and matches it at the small scale under equal training budgets. Furthermore, under the same training time, *Pairmixer* exceeds the sequence-only Transformer baseline across all scales. These results suggest that a sequence-only Transformer is inadequate for extracting structural features, while the triangle multiplications and feed-forward networks in *Pairmixer* are sufficient to capture rich structural representations. Full tabular results are provided in Table 4 and Table 5, and detailed FLOPs analysis is provided in Section B.

5.3 Inference time comparisons

Many downstream applications require running the structure predictor on thousands to millions of complexes, making inference efficiency critical. In Figure 4, we benchmark *Pairmixer* against the Pairformer and a sequence-only transformer under a default setting of 512 tokens, 4608 atoms, MSA depth of 4096, 10 recycles, 48 blocks, and 200 sampling steps.

On this setup, Boltz-1 requires 34 seconds to generate a single sample on a GH200 GPU, while *Pairmixer* completes in 21 seconds, yielding a 1.6× speedup. This advantage holds consistently across different recycle counts, MSA depths, and backbone sizes. The scaling benefits are even more striking for longer sequences: at 1024 tokens, *Pairmixer* is 2× faster, and at 2048 tokens, it delivers a 4× speedup, reducing runtime from 1000 seconds to 250 seconds. These results establish *Pairmixer* as a scalable and efficient architecture, making large-scale cofolding more practical.

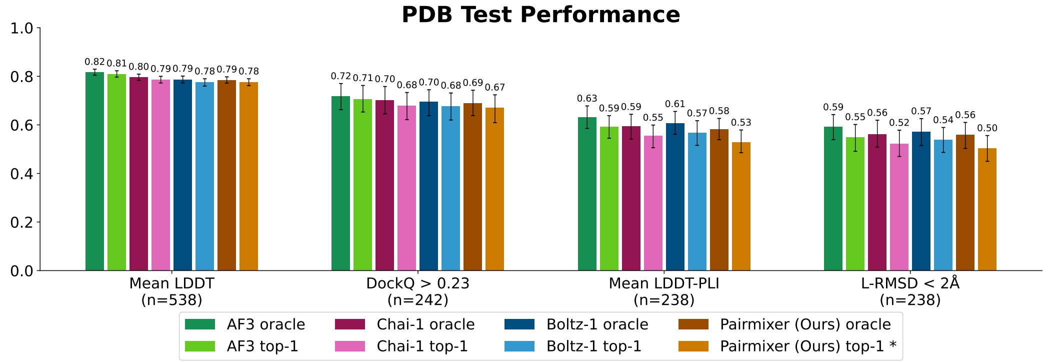


Figure 6: **System-level comparison on the RCSB test set.** We evaluate against AlphaFold3, Chai-1, and Boltz-1 on protein and small-molecule structure prediction. *Pairmixer* performs competitively with these state-of-the-art approaches. Error bars denote bootstrapped 95% confidence intervals. *Since we do not train a confidence model, results are reported using the first prediction.

Table 1: **Runtime comparison of generating proteins with *Pairmixer* and Pairformer in the BoltzDesign framework.** For biologically relevant targets of various sequence lengths, we generate three 110-residue binders using 160 iterations in all settings and report the average running time.

Target	PDB_Chain	Complex	Target	Pairmixer	Pairformer	Speedup
		Length	Length	Time (sec)	Time (sec)	
GIP peptide	2QHK_B	140	30	680.0	337.6	2.01×
Ubiquitin	1UBQ_A	186	76	1113.6	532.8	2.09×
TP53	4MZI_A	303	193	3198.4	1390.4	2.30×
hSDH	1P5J_A	429	319	7289.6	2920	2.50×
hMAO	1GOS_A	607	497	17134.4	6601.6	2.60×
bsDNA Polymerase	3TAN_A	702	592	9184	OOM	∞
hTLR3	1ZIW_A	739	629	10568	OOM	∞
Prostate Antigen (PSA)	1Z8L_A	805	695	OOM	OOM	–

5.4 Comparisons to prior works

Figure 6 compares *Pairmixer* to other cofolding models on the RCSB test set, evaluating protein folding, protein–protein interactions (DockQ), and protein–ligand interactions (lDDT-PLI and ligand RMSD < 2). See Section 5.2 for a description of the test dataset. We generate five poses per complex and report both the performance of the best pose (oracle) and the average across poses. Results for existing methods are taken from the literature. *Pairmixer* matches Boltz-1 in mean lDDT and protein–ligand lDDT, slightly improves ligand RMSD < 2 (0.55 vs. 0.54), but lags on DockQ > 0.23 (0.63 vs. 0.64). These results indicate that even at the largest scale, triangle multiplication and pair FFNs in *Pairmixer* are sufficient for cofolding across diverse interaction types. We show similar results on the CASP15 test set in Section C.2.

5.5 Comparisons on binder design (BindFast)

BindCraft (Pacesa et al., 2025), BoltzDesign (Cho et al., 2025), and hallucination-based protein design methods (Frank et al., 2024; Wicky et al., 2022; Jendrusch et al., 2025; Goverde et al., 2023; Bryant & Elofsson, 2022; Anishchenko et al., 2021) have recently demonstrated that structure predictors can be repurposed as differentiable scoring functions for sequence optimization. The input sequence is treated as a set of learnable parameters and is updated by backpropagating through a structure predictor, thereby jointly refining sequence and structure toward favorable interactions with the target protein or small molecule. While powerful, these methods have practical limitations: memory demands are high and sequence generation is slow, requiring hundreds of runs of the structure predictor per design. This inefficiency makes the approach prohibitively expensive, particularly for larger systems.

To address these challenges, we introduce BindFast, a scalable and efficient framework for binder design which replaces BoltzDesign's Pairformer backbone with *Pairmixer*. BindFast substantially reduces the runtime and memory footprint of binder generation and aim to accelerate the discovery of high-quality binders, particularly for large targets.

In Table 1, we benchmark the runtime performance of BindFast against BoltzDesign for generating 110-residue binders across a range of target proteins with biotechnological relevance, using an A100 GPU with 80 GB memory. BoltzDesign failed with out-of-memory (OOM) errors on targets larger than 500 residues, whereas BindFast extended this limit to 650 residues, a 30% improvement in target size. For protein targets where both models executed without memory overflow, BindFast achieves speedups of 2x to 2.6x at total sequence lengths ranging from 140 to 607, respectively. Qualitative comparisons in Figure 13 further indicate that BindFast produces designs comparable to those of BoltzDesign, underscoring its potential for faster in-silico iteration and enabling the design of binders against larger, more biologically relevant targets.

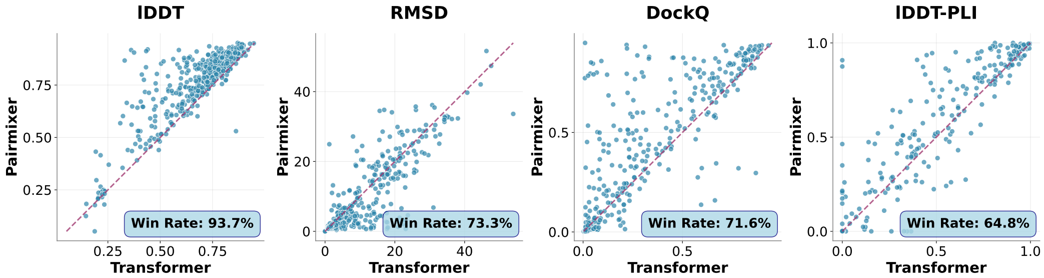


Figure 7: **Head-to-head comparison between *Pairmixer* and the Transformer backbone.** The win rate shows how often the *Pairmixer* architecture achieves a better score than the Transformer architecture. *Pairmixer* outperforms the Transformer on the distance-based lDDT metric in 93.7% of the cases, highlighting that its advantage lies in capturing pairwise interactions.

6 Analysis

Predicting biomolecular structure requires reasoning over the entire sequence to capture diverse interactions among residues. We analyze how the architectural design of modern structure predictors facilitates such reasoning and how the simplified *Pairmixer* architecture achieves this.

Pair representations. A central challenge in biomolecular structure prediction is determining the strength of the interactions between all residue pairs. This is difficult because folding involves nonlocal tertiary interactions in which residues distant in sequence often interact physically in three-dimensional space. Modern structure predictors address this challenge with a pair representation. Our results indicate that the pair representation enables the model to capture fine-grain spatial relationships between all residue pairs.

We compare the performance of *Pairmixer*, which incorporates pair representations, against our sequence-only Transformer baseline in Figure 7. On the lDDT metric computed from pairwise distances, *Pairmixer* achieves higher scores in 93.7% of test complexes. In contrast, on the RMSD metric, which requires global structural alignment, the improvement is smaller (74.7%). These findings show that pair representations provide greater benefits for local, pairwise accuracy over sequence attention, suggesting their effectiveness in capturing residue–residue interactions.

Triangle multiplication. Modern structure predictors employ triangle attention and triangle multiplication within the pair representation to capture geometric relationships among residue triplets. While triangle attention allows the model to reason *sparsely* over interacting residues, triangle multiplication *densely* aggregates features across the entire sequence. However, our analysis shows that triangle multiplication also efficiently captures sparse geometric relationships among residue triplets by adjusting the magnitudes in the pair representations.

We explicitly sparsify triangle multiplication by introducing dropout during inference. Formally,

$$\mathbf{TriMulWithDropout}(\boldsymbol{z})_{ij} = \sum_{k=1}^{L} (\boldsymbol{W}_a \boldsymbol{z}_{ik}) \odot (\boldsymbol{W}_b \boldsymbol{z}_{jk}) \cdot \underbrace{M(\boldsymbol{z}_{ik})\, M(\boldsymbol{z}_{jk})}_{\text{new dropout masks}}$$

where $M(\boldsymbol{z}_{ij}) \in \{0, 1\}$ determines whether a particular interaction is active. In *random dropout* with dropout rate $\gamma \in [0, 1]$, the masks are sampled independently as $M(\boldsymbol{z}_{ik}), M(\boldsymbol{z}_{jk}) \sim \text{Bernoulli}(1-\gamma)$. Each term is retained only if both corresponding masks are active, resulting in a higher effective dropout rate.

We experiment with a *low-norm dropout* scheme, where the probability of retaining an interaction (i, j) depends on the magnitude of its pair representation $\|\boldsymbol{z}_{ij}\|$. With dropout rate $\gamma \in [0, 1]$,

$$M(\boldsymbol{z}_{ik}) = \begin{cases} 1, & \text{if } k \in \text{Top}_{1-\gamma}(\{\|\boldsymbol{z}_{il}\|\}_{l=1}^{L}) \\ 0, & \text{otherwise.} \end{cases}$$

where the features with the γ smallest magnitudes are dropped out.

Figure 8 shows the performance of the model where both dropout schemes are applied to every layer with $\gamma = 0, 0.10, 0.25, 0.50, 0.75$. We observe that performance starts to degrade rapidly once

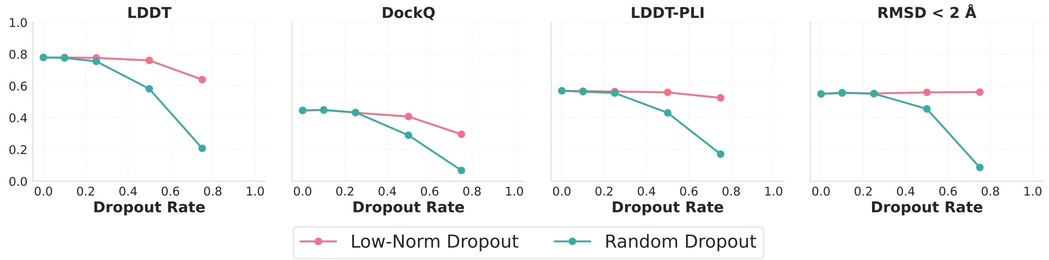


Figure 8: ***Pairmixer* Performance under different sparse triangle multiplication variants.** The model is trained with standard triangle multiplication and evaluated under various dropout conditions. While performance degrades rapidly under random dropout, it remains stable when low-norm entries in the triangle multiplication are zeroed out.

the random dropout rate exceeds 25%, indicating that the model is not robust to random removal of interactions. However, the performance is very similar under the low-norm dropout of 75%. This suggests that, like attention, triangle multiplication identifies and processes a small subset of interactions that are essential for accurate folding of biomolecular complexes.

Additionally, we aim to disentangle the role of the pair representation, which scales quadratically with sequence length, from the cubic computational cost of triangle operations. To isolate these effects, we study the analogous Match3 problem, a standard representation learning benchmark in which the model predicts whether any three elements in a set sum to zero (Sanford et al., 2023; Kozachinskiy et al., 2025) (see Section C.5 for details). We compare three architectures: standard self-attention (linear representations, quadratic compute), third-order self-attention (linear representations, cubic compute as in Roy et al. (2025)), and triangle multiplication (quadratic representations, cubic compute). For shallow architectures, triangle multiplication on the pair representation outperforms both Transformer variants, highlighting its ability to capture nonlocal 3-token interactions using quadratic pair representations (see Figure 14).

7 Conclusion

We introduce *Pairmixer*, a simplified, efficient feature extractor for biomolecular structure prediction. Models using *Pairmixer* train 1.5× faster and sample up to 4× faster than those with Pairformer, enabling large-scale, compute-intensive applications of structure prediction. The key idea is to explicitly materialize a 2-D pair representation, updated via triangle multiplications that capture interactions among residue triplets. We hypothesize that transforming 1-D sequences into 3-D structures is most effective when mediated through this intermediate pair representation, which naturally encodes distance information. Triangle multiplication provides a simple and efficient mechanism to do so.

8 Acknowledgment

This work was in part supported by the NSF AI Institute for Foundations of Machine Learning (IFML) and UT-Austin Center for Generative AI.

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. 1, 2, 4, 6

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. 2

Yo Akiyama, Zhidian Zhang, Milot Mirdita, Martin Steinegger, and Sergey Ovchinnikov. Scaling down protein language modeling with msa pairformer. *bioRxiv*, pp. 2025–08, 2025. 3

Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021. 8

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. 2

Minkyung Baek, Ivan Anishchenko, Ian R Humphreys, Qian Cong, David Baker, and Frank DiMaio. Efficient and accurate prediction of protein structure using rosettafold2. *BioRxiv*, pp. 2023–05, 2023. 2

Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature methods*, 21(1): 117–121, 2024. 2

Steven A Benner and Dietlinde Gerloff. Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. *Advances in Enzyme Regulation*, 31:121–181, 1991. 4

Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life. *BioRxiv*, 2024. 2

Patrick Bryant and Arne Elofsson. Evobind: in silico directed evolution of peptide binders with alphafold. *bioRxiv*, pp. 2022–07, 2022. 8

ByteDance, Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, Shenghao Wu, Kuangqi Zhou, Yanping Yang, Zhenyu Liu, Lan Wang, Bo Shi, Shaochen Shi, and Wenzhi Xiao. Protenix - advancing structure prediction through a comprehensive alphafold3 reproduction. *bioRxiv*, 2025. doi: 10.1101/2025.01.08.631967. URL https://www.biorxiv.org/content/early/2025/01/11/2025.01.08.631967. 2

Yehlin Cho, Martin Pacesa, Zhidian Zhang, Bruno E Correia, and Sergey Ovchinnikov. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *bioRxiv*, pp. 2025–04, 2025. 2, 8

Giannis Daras, Jeffrey Ouyang-Zhang, Krithika Ravishankar, William Daspit, Costis Daskalakis, Qiang Liu, Adam Klivans, and Daniel J Diaz. Ambient proteins: Training diffusion models on low quality structures. *bioRxiv*, pp. 2025–07, 2025. 2

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021. 2

Christopher Frank, Ali Khoshouei, Lara Fuβ, Dominik Schiwietz, Dominik Putz, Lara Weber, Zhixuan Zhao, Motoyuki Hattori, Shihao Feng, Yosta de Stigter, et al. Scalable protein design using optimization in a relaxed sequence space. *Science*, 386(6720):439–445, 2024. 8

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025. 2

Casper A Goverde, Benedict Wolf, Hamed Khakzad, Stéphane Rosset, and Bruno E Correia. De novo protein design by inversion of the alphafold structure prediction network. *Protein Science*, 32 (6):e4653, 2023. 8

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. 2

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, 11 2024. ISSN 2631-9268. doi: 10.1093/nargab/lqae150. URL https://doi.org/10.1093/nargab/lqae150. 2

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. 18

Alexander M Ille, Christopher Markosian, Stephen K Burley, Renata Pasqualini, and Wadih Arap. Human protein interactome structure prediction at scale with boltz-2. *bioRxiv*, pp. 2025–07, 2025. 3

IntFold, Leon Qiao, Wayne Bai, He Yan, Gary Liu, Nova Xi, Xiang Zhang, and Siqi Sun. Intfold: A controllable foundation model for general and specialized biomolecular structure prediction. *arXiv preprint arXiv:2507.02025*, 2025. 2

Michael A Jendrusch, Alessio LJ Yang, Elisabetta Cacace, Jacob Bobonis, Carlos GP Voogdt, Sarah Kaspar, Kristian Schweimer, Cecilia Perez-Borrajero, Karine Lapouge, Jacob Scheurich, et al. Alphadesign: A de novo protein design framework based on alphafold. *Molecular Systems Biology*, pp. 1–24, 2025. 8

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. 1, 2, 3, 6, 16

Alexander Kozachinskiy, Felipe Urrutia, Hector Jimenez, Tomasz Steifer, Germán Pizarro, Matías Fuentes, Francisco Meza, Cristian B. Calderon, and Cristóbal Rojas. Strassen attention: Unlocking compositional abilities in transformers based on a new lower bound method. *arXiv*, 2025. doi: 10.48550/ARXIV.2501.19215. URL https://arxiv.org/abs/2501.19215. 10

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021. 3, 19

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed 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. 2, 3

Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011. 4

Jeffrey Ouyang-Zhang, Chengyue Gong, Yue Zhao, Philipp Krähenbühl, Adam R Klivans, and Daniel J Diaz. Distilling structural representations into protein sequence models. *bioRxiv*, pp. 2024–11, 2024. 2

Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *elife*, 3: e02030, 2014. 4

Sergey Ovchinnikov, Hahnbeom Park, Neha Varghese, Po-Ssu Huang, Georgios A Pavlopoulos, David E Kim, Hetunandan Kamisetty, Nikos C Kyrpides, and David Baker. Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298, 2017. 4

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. One-shot design of functional protein binders with bindcraft. *Nature*, pp. 1–10, 2025. 2, 3, 8

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. *arXiv*, 2023. doi: 10.48550/ARXIV.2302.10866. URL https://arxiv.org/abs/2302.10866. 3

Aurko Roy, Timothy Chou, Sai Surya Duvvuri, Sijia Chen, Jiecao Yu, Xiaodong Wang, Manzil Zaheer, and Rohan Anil. Fast and simplex: 2-simplicial attention in triton. *arXiv*, 2025. doi: 10.48550/ARXIV.2507.02754. URL https://arxiv.org/abs/2507.02754. 10

Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications*, 14(1):2389, 2023. 3

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *37th Conference on Neural Information Processing Systems*, 2023. doi: 10.48550/ARXIV.2306.02896. URL https://arxiv.org/abs/2306.02896. 10

Valeria Scardino, Juan I Di Filippo, and Claudio N Cavasotto. How good are alphafold models for docking-based virtual screening? *Iscience*, 26(1), 2023. 2

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020. 1, 2

Yoav Shamir and Nir London. State-of-the-art covalent virtual screening with alphafold3. *bioRxiv*, pp. 2025–03, 2025. 2

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 3

Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024. 2

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022. 2

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Josh Susskind, and Miguel Angel Bautista. Simplefold: Folding proteins is simpler than you think. *arXiv preprint arXiv:2509.18480*, 2025. 2

Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009. 4

Basile IM Wicky, Lukas F Milles, Alexis Courbet, Robert J Ragotte, Justas Dauparas, E Kinfu, S Tipps, Ryan D Kibler, Minkyung Baek, Frank DiMaio, et al. Hallucinating symmetric protein assemblies. *Science*, 378(6615):56–61, 2022. 8

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, 2024. doi: 10.1101/2024.11.19.624167. 1, 2, 6, 18

Jeremy Wohlwend, Mateo Reveiz, Matt McPartlon, Axel Feldmann, Wengong Jin, and Regina Barzilay. Minifold: Simple, fast, and accurate protein structure prediction. *Transactions on Machine Learning Research*, 2025. 2

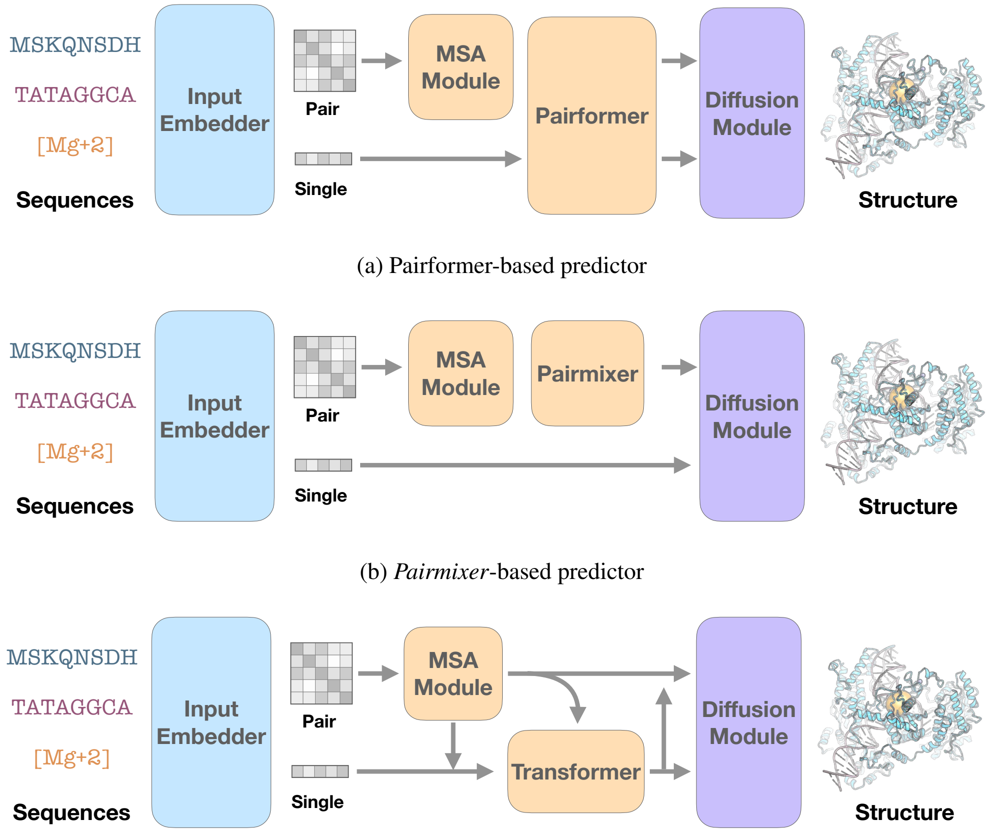
Felix Wong, Aarti Krishnan, Erica J Zheng, Hannes Stärk, Abigail L Manson, Ashlee M Earl, Tommi Jaakkola, and James J Collins. Benchmarking alphafold-enabled molecular docking predictions for antibiotic discovery. *Molecular systems biology*, 18(9):e11081, 2022. 2

Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020. 2

Charles Yanofsky, Virginia Horn, and Deanna Thorpe. Protein structure relationships revealed by mutational analysis. *Science*, 146(3651):1593–1594, 1964. 4

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv*, 2021. doi: 10.48550/ARXIV.2105.14103. URL https://arxiv.org/abs/2105.14103. 3

Jing Zhang, Ian R Humphreys, Jimin Pei, Jinuk Kim, Chulwon Choi, Rongqing Yuan, Jesse Durham, Siqi Liu, Hee-Jung Choi, Minkyung Baek, et al. Predicting protein-protein interactions in the human proteome. *Science*, pp. eadt1630, 2025. 3



(a) Pairformer-based predictor

(b) *Pairmixer*-based predictor

(c) Transformer-based predictor

Figure 9: **Overview of biomolecular structure predictors.** We study the effect of varying backbone architectures while keeping all other modules fixed, except in the Transformer model, where we adjust the connections between the MSA module outputs and the Diffusion module inputs.

A Architectural Baselines

The full cofolding pipeline for all methods can be found at Figure 9.

A.1 Pairformer Baseline

Here we describe the Pairformer architecture of Figure 10 in detail.

Attention Primitive. The Pairformer extends the standard attention mechanism by incorporating a pairwise bias term derived from the pair representation $\boldsymbol{z}$. Formally, this update is

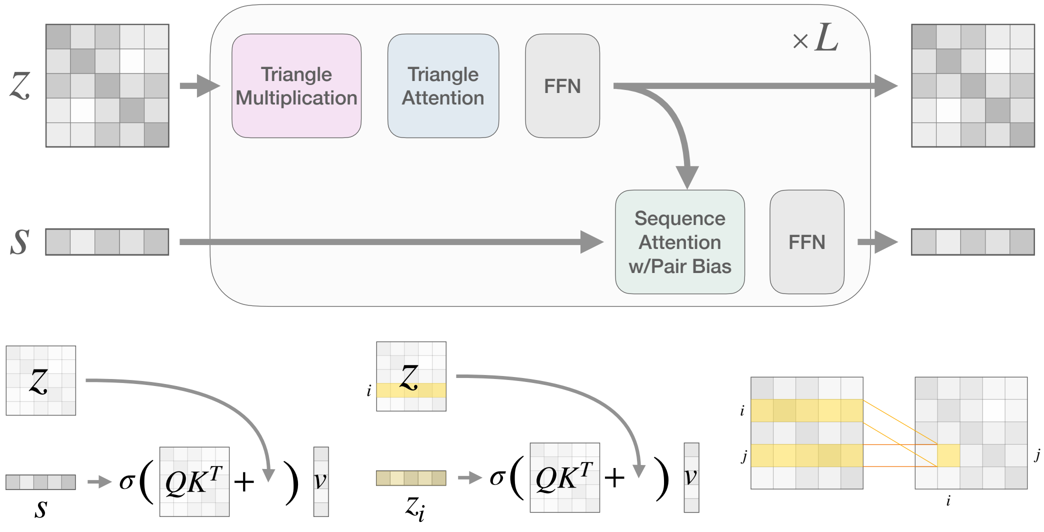
$$\mathbf{AttnWithPairBias}(\boldsymbol{x}, \boldsymbol{z}) = \text{softmax}\Big((\boldsymbol{W}_Q\boldsymbol{x})(\boldsymbol{W}_K\boldsymbol{x})^\top + \boldsymbol{W}_B\boldsymbol{z}\Big)\,\boldsymbol{W}_V\boldsymbol{x},$$

where $\boldsymbol{x} \in \mathbb{R}^{L\times C_x}$ is a sequence representation, $\boldsymbol{z} \in \mathbb{R}^{L\times L\times C_z}$ is a pair representation, $(\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V)$ are standard attention projection matrices, and $\boldsymbol{W}_B$ projects the pair representation into an attention bias term [2].

The Sequence Update first performs attention with pair bias (see Figure 10a) and then applies a feed-forward network. At layer l, we compute the update

$$\tilde{\boldsymbol{s}}_{l+1} = \boldsymbol{s}_l + \mathbf{AttnWithPairBias}(\boldsymbol{s}_l, \boldsymbol{z}_l)$$
$$\boldsymbol{s}_{l+1} = \tilde{\boldsymbol{s}}_{l+1} + \mathbf{FFN}(\tilde{\boldsymbol{s}}_{l+1})$$

[2] single head and removed scaling for brevity



(a) Seq. Attention w/Pair Bias (b) Triangle Attention (c) Triangle Multiplication

Figure 10: **Pairformer Architecture and Module Details.** The main architecture (top) outlines the general Pairformer layer. The detailed module architectures (bottom) illustrate the key components: (a) Sequence Attention with Pair Bias, (b) Triangle Attention, and (c) Triangle Multiplication modules.

The Pair Update mixes the tokens in pair representation $\boldsymbol{z} \in \mathbb{R}^{L\times L\times C_z}$ using triangle attention and triangle multiplication, then applies a feedforward network.

The *Triangle Attention* operates on each row of the pair representation $\boldsymbol{z}_i \in \mathbb{R}^{L\times C_z}$ as an independent sequence, applying sequence attention with pair bias to each row separately[3] (see Figure 10b). Formally, the update for row i is defined as

$$\mathbf{TriAttn}(\boldsymbol{z})_i = \mathbf{AttnWithPairBias}(\boldsymbol{z}_i, \boldsymbol{z})$$

The *Triangle Multiplication* integrates features across different rows of the pair representation [4] (see Figure 10c). Formally, the update for feature $\boldsymbol{z}_{ij}$ is defined as

$$\mathbf{TriMul}(\boldsymbol{z})_{ij} = \sum_{k=1}^{L} (\boldsymbol{W}_a \boldsymbol{z}_{ik}) \odot (\boldsymbol{W}_b \boldsymbol{z}_{jk})$$

where $\boldsymbol{W}_a, \boldsymbol{W}_b$ are linear projection layers.

Both pair operations were introduced to reason over triplets of residues, intuitively enabling the model to learn to follow geometric constraints in 3-D space (Jumper et al., 2021).

A.2 Transformer Baseline

Our transformer baseline removes the pair update from the Pairformer and keeps only the sequence update. We also modify the MSA module to make it more effective with the transformer baseline. Instead of outputting only $\boldsymbol{z}^{\text{msa}}$, it produces an additional sequence representation $\boldsymbol{s}^{\text{msa}}$, obtained by indexing the first row of the processed MSA representation. This $\boldsymbol{s}^{\text{msa}}$ is fed into the transformer, while $\boldsymbol{z}^{\text{msa}}$ serves as the pair bias. Additionally, the diffusion module expects both sequence and pair representations. Because the pair features are otherwise less processed in this baseline, we update them with the outer sum of the sequence representation. Formally,

$$\boldsymbol{z}_{ij}^{\text{backbone}} = \boldsymbol{z}_{ij}^{\text{msa}} + \boldsymbol{W}_{\boldsymbol{s}\to\boldsymbol{z}} \boldsymbol{s}_i^{\text{backbone}} + \boldsymbol{W}_{\boldsymbol{s}\to\boldsymbol{z}} \boldsymbol{s}_j^{\text{backbone}}$$

where $\boldsymbol{W}_{\boldsymbol{s}\to\boldsymbol{z}} \in \mathbb{R}^{C_z\times C_s}$ is a projection layer. This is illustrated in Figure 9c.

[3] In practice, another layer of triangle attention is performed on the columns.

[4] In practice, another layer of triangle multiplication is performed on the columns.

Module / Operation	**FLOPs**
Backbone / MSA Module	
Pair Update	
Triangle Attention	
Matrix Multiply	$8\,L^3C_z$
Projection	$20\,L^2C_z^2$
Triangle Multiplication	
EinSum	$4\,L^3C_z$
Projection	$24\,L^2C_z^2$
Pair FFN	$24\,L^2C_z^2$
Sequence Update	
Sequence Attention (with Pair Bias)	
Matrix Multiply	$4\,L^2C_s$
Projection	$10\,LC_s^2$
Sequence FFN	$24\,LC_s^2$
Diffusion Transformer	
Attention (Pair Bias) – Matrix Multiply	$4\,L^2C_a$
Attention (Pair Bias) – Projection	$10\,LC_a^2$
Sequence FFN	$16\,LC_a^2$
Full Modules	
MSA Module	$R\,D_m\,(12L^3C_z + 68L^2C_z^2)$
Pairformer	$R\,D_p\,(12L^3C_z + 68L^2C_z^2 + 4L^2C_s + 34LC_s^2)$
Structure Module	$M\,D_d\,(4L^2C_a + 26LC_a^2)$

Table 2: **Breakdown of FLOPs in *AlphaFold3* architectural components.** Variables: L = `max_tokens`, C_z = `token_z`, C_s = `token_s`, $C_a = 2 \times$ `token_z`, R = `recycles`, D_p = `pairformer_depth`, D_m = `msa_depth`, D_d = `diffusion_depth`, M = `multiplicity`.

B FLOPs Calculations

Our biomolecular structure predictor uses a multi-resolution transformer that denoises atom coordinates at both the token and heavy-atom levels (see Figure 2). In this design, a backbone refines token representations, which are then processed by a conditional diffusion transformer. The backbone runs once per sequence, while the diffusion transformer can generate arbitrarily many samples.

In Table 2, we present the mathematical FLOP calculations for each component, and in Table 6 we report the total training and inference FLOPs for all model architectures.

Boltz-1 Hyperparameters The Boltz-1 architecture is defined by several key components and hyperparameters that influence its performance. We identify the following set of critical hyperparameters:

- **Input**: The input is defined by the number of input tokens (L), the single token dimension (C_s), and the pair token dimension (C_z).
- **Feature extractor**: The feature extractor consists of Pairformer and MSA blocks that process single and pair representations; its configuration is determined by the number of Pairformer blocks D_p, MSA blocks D_m.
- **Diffusion model**: The diffusion model is a transformer architecture made up of Multi-Head Attention (MHA) transformer layers. Its configuration is determined by the number of diffusion blocks (D_d) and the widths of its layers $C_a = 2\,C_z$.

Feature extractors. The feature extractors is a concatenation of D_m MSA blocks and D_p pairformer blocks. Each pairformer block primarily consists of two parallel update paths: the pair representation path and the single representation path (see Figure 10). Each path is further processed by a FFN. The pair representation path includes two *triangular self-attention* updates and two *triangular multiplication* updates (applied row-wise and column-wise). These are analogous to axial attention

Table 3: ***Pairmixer* ablations experiments.** Default settings are marked in grey. See Section C.1 for details. D_p: number of pairmixer layers. D_d: number of diffusion transformer layers.

(a) **FFN Hidden Dimension**

dim	lDDT	DOCKQ$_{>0.49}$	lDDT$_{\text{PLI}}$	RMSD$_{<1}$
256	0.71	0.38	0.50	0.34
512	0.71	**0.42**	0.50	0.33
1024	**0.74**	0.40	**0.53**	**0.35**

(b) **Triangle Mul Dimension**

dim	lDDT	DOCKQ$_{>0.49}$	lDDT$_{\text{PLI}}$	RMSD$_{<1}$
64	0.71	0.41	0.50	0.34
128	0.71	**0.42**	0.50	0.33
256	**0.73**	**0.42**	**0.52**	**0.37**

(c) **Mixing Method**

mixer	lDDT	DOCKQ$_{>0.49}$	lDDT$_{\text{PLI}}$	RMSD$_{<1}$
FFT	0.66	0.34	0.45	0.27
AvgPool	0.69	0.35	0.48	0.31
TriMul-rows	0.70	0.35	0.49	0.32
TriMul-both	**0.71**	**0.42**	**0.50**	**0.33**

(d) **Diffusion Transformer Depth**

D_p	D_d	lDDT	DOCKQ$_{>0.49}$	lDDT$_{\text{PLI}}$	RMSD$_{<1}$
12	12	0.73	0.43	0.52	0.34
24	24	**0.75**	**0.45**	**0.54**	**0.40**

mechanisms (Ho et al., 2019) operating over an $L \times L$ pair matrix, where each attention pass involves computations along one length-L dimension for each of the L rows or columns.

Each pair of triangular attention pass incurs a computational cost of $O(8L^3C_z)$ FLOPs. The triangle multiplication einsum operations require a quadratic FLOPs term per input token (total FLOPs of $4L^3C_z$). Following the triangle updates, a feed-forward network (FFN) is applied to each pair representation entry. The single representation path also contributes to the computational load, but its cost is quadratic in L.

Each MSA block is lighter than the full pairformer blocks and consists of a pair of triangular attention layers and a pair of triangular operations, followed by a FFN network for pair representation FFN (C_z), but without a single representation FFN and attention with pair bias. It also includes an additional OuterProductMean and pair-weighted averaging on the MSA, which we omit from our FLOPs calculations.

Diffusion Model. Each diffusion module block resembles a standard transformer block with a standard *self-attention* mechanism and a conditioning block. As with the trunk block analysis, we ignore bias terms, gating, and layer normalization for simplicity. We also ignore the cost of Atom Attention Encoder and Atom Attention Decoder that run on atoms, since those modules adopt sequence-local attention (Wohlwend et al., 2024) and their computational cost is negligible. The conditioned transition block of the diffusion model is dominated by dense matrix multiplications that scale quadratically with the hidden size C_a. The bulk of the compute arises from the SwiGLU feed-forward pathway, which contributes both a pair of linear projections $(4\,C_a^2)$ and the associated activation matmul $(2\,C_a^2)$. In addition, cross–path transformations are introduced via the $a \to b$ and $b \to a$ projections (each $2\,C_a^2$), followed by an output projection $(2\,C_a^2)$. Finally, the gating mechanisms for both the a and b streams contribute another $2\,C_a^2$ apiece. The total FLOPs per structure block can therefore be approximated as the sum of the attention, MatMuls, and feed-forward components (see Table 2).

C Results

C.1 Ablations

Triangle Multiplication vs. Feed-Forward Network. We aim to understand how the performance is affected by the triangle multiplication and pair feed-forward networks, the two core ingredients of the *Pairmixer* architecture. In Table 3a and Table 3b, we vary the hidden dimensions of these components to evaluate model's sensitivity. For the FFN, we change the hidden dimension that the model expands to. For triangle multiplication, we instead project the features into higher- or lower-dimensional spaces before the multiplication and then project them back to the input dimension. We find that decreasing the FFN hidden dimension does not change performance much, while doubling the FFN dimension increases the mean lDDT from 0.71 to 0.74. We see a similar trend with triangle multiplication dimensions – doubling the hidden dimension improves the mean lDDT from 0.71 to 0.73, while reducing the dimensionality does not change lDDT.

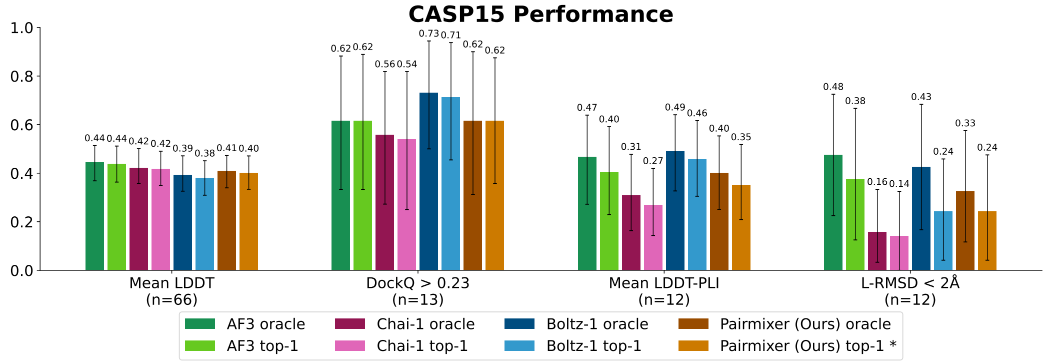


Figure 11: **System-level comparison on the CASP15 test set.** We evaluate against AlphaFold3, Chai-1, and Boltz-1 on protein and small-molecule structure prediction. *Pairmixer* performs competitively with these state-of-the-art approaches. Error bars denote bootstrapped 95% confidence intervals. *Since we do not train a confidence model, results are reported using the first prediction.

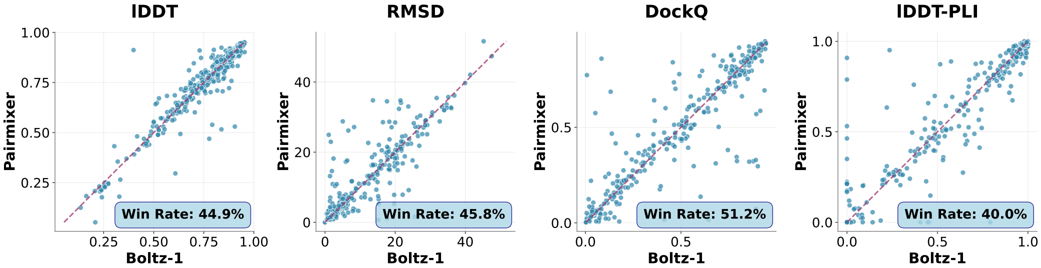


Figure 12: **Head-to-head comparison between *Pairmixer* and the Pairformer backbone.** The win rate shows how often the *Pairmixer* architecture achieves a better score than the Transformer architecture. In 89% of cases, the two models' lDDT scores differ by less than 5 points.

Other mixing methods. Triangle multiplication mixes features within the $z \in \mathbb{R}^{L \times L \times D}$ pair representation. In Table 3c, we replace this operation with alternative, simpler mixing functions. First, we ablate the outgoing triangle multiplications, retaining only the incoming variant. Second, we introduce an FFT mixer that applies the discrete Fourier transform along rows and columns, following FNet (Lee-Thorp et al., 2021). Finally, we test a pooling mixer that averages representations across each row (and column) and adds the result back to all positions along the corresponding axis.

We find that these simplified approaches are insufficient and underperform compared to vanilla triangle multiplication. For instance, the FFT mixer likely fails because it mixes features solely based on sequence position, ignoring discontinuities introduced by multiple chains.

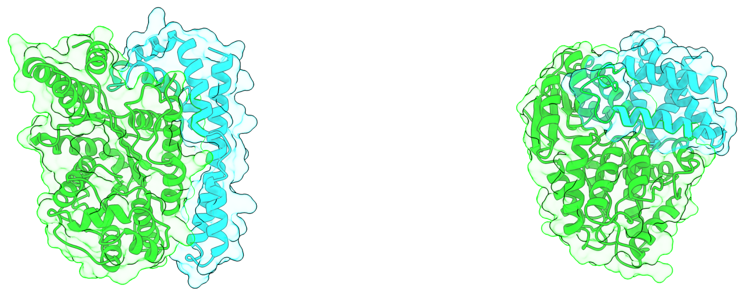
Diffusion Module. The diffusion module takes the latent representations as input and decodes the 3-dimensional protein structure using a 24-layer transformer. In Table 3, we evaluate how sensitive the Pairformer and *Pairmixer* architectures are to the size of the diffusion module.

C.2 System-level comparisons on the CASP15 dataset

We report results on the CASP15 dataset in Table 11. These numbers differ slightly from Table 5 because we further filter proteins to ensure all methods are evaluated on the same set.

C.3 Head-to-head comparisons between Pairmixer and Pairformer

In Figure 12, we show the performance of *Pairmixer* and Pairformer on the Boltz RCSB test set. We find that the model's performance is fairly correlated, indicating that the models have learned similar representations.



(a) Pairformer-based predictions (b) *Pairmixer*-based predictions

Figure 13: **Qualitative de-novo binders. PDB code: 1P5J** Target is shown in green and binder is shown in blue.

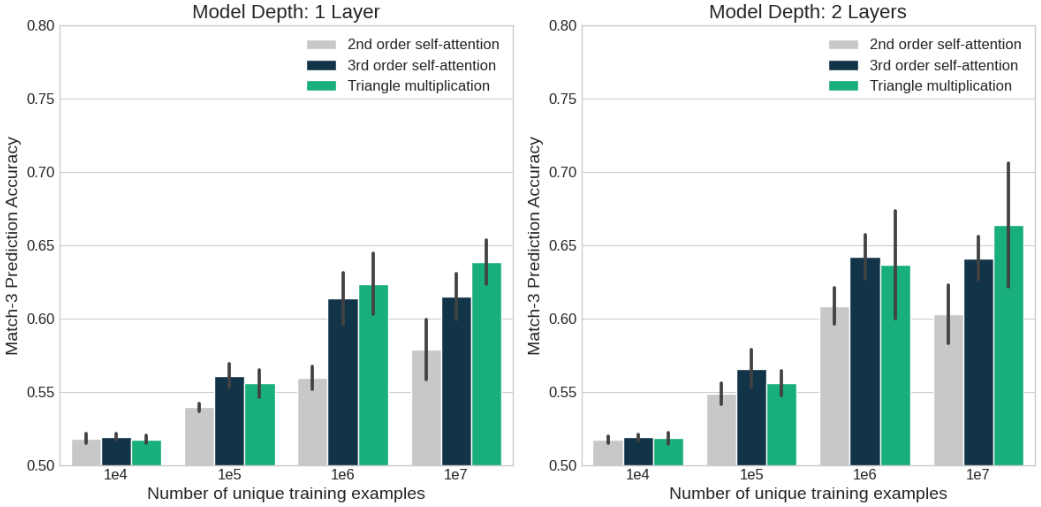


Figure 14: **Comparison between architecture variants on Match3.** We report classification accuracy on Match3 task as a function of training data size and model depth.

C.4 Qualitative visualization of de-novo binders

Figure 13 shows qualitative comparisons between BindFast and BotlzDesign, showing that they produce comparable designs.

C.5 Match3 Task

Match3 is a binary classification task, in which the model is given a sequence $\mathbf{x} \in [M]^N$ for sequence length N and positive integer M and predicts whether the sequence contains a triple $\{x_i, x_j, x_k\}$ for i, j, k distinct such that $x_i + x_j + x_k \equiv 0 \mod M$. We set $N = 16$ and $M = 64$. Models use a hidden dimension of size 8 with an embedding and final projection layers to map representations to the appropriate dimensionality. This ensures that parameter count between architecture types is comparable, varying from 900 for the 2nd-order attention layer to 1100 for the triangle multiplication layer. We max-pool over tokens to construct the model's final classification logits. Each model is trained and tested on a balanced set of positives and negatives for this task.

We observe that all model architectures do poorly on highly data-constrained regimes, but as data and depth increase, the standard transformer architecture lags behind in performance (see Figure 14). The triangle multiplication layer is effective at learning this task quickly, outpacing the 3rd-order attention methods, which have the same asymptotic computational complexity with respect to sequence length.

Table 4: **Model Performance on the Boltz RCSB test set.** The metric is computed on the best-performing protein out of five samples (oracle).

Architecture	Epoch	GPU-Days	lDDT (n=539)	DOCKQ$_{>0.23}$ (n=342)	DOCKQ$_{>0.49}$ (n=342)	lDDT$_{\text{PLI}}$ (n=250)	RMSD$_{<1}$ (n=250)	RMSD$_{<2}$ (n=250)
Small								
Transformer	68	86	0.68	0.51	0.35	0.47	0.32	0.43
Pairformer (Boltz-1)	68	125	0.74	0.58	0.44	0.52	0.37	0.48
Pairmixer (Ours)	68	98	0.73	0.59	0.44	0.51	0.33	0.45
Medium								
Transformer	68	128	0.67	0.50	0.36	0.47	0.33	0.46
Pairformer (Boltz-1)	68	194	0.75	0.60	0.47	0.53	0.36	0.49
Pairmixer (Ours)	68	146	0.76	0.60	0.46	0.54	0.40	0.53
Large								
Transformer	68	173	0.69	0.51	0.37	0.48	0.33	0.46
Pairformer (Boltz-1)	68	290	0.76	0.61	0.49	0.54	0.41	0.52
Pairmixer (Ours)	68	192	0.75	0.61	0.46	0.55	0.38	0.51
Large Phase 2								
Transformer	20	232	0.70	0.53	0.38	0.51	0.35	0.48
Pairformer (Boltz-1)	20	421	0.78	0.64	0.50	0.57	0.44	0.54
Pairmixer (Ours)	20	269	0.78	0.63	0.49	0.57	0.45	0.55
Boltz-1 public model								
Pairformer (Boltz-1)	-	-	0.79	0.64	0.51	0.58	0.46	0.57

Table 5: **Model Performance on CASP15 test set.** The metric is computed on the best-performing protein out of five samples (oracle).

Architecture	Epoch	GPU-Days	lDDT (n=66)	DOCKQ$_{>0.23}$ (n=14)	DOCKQ$_{>0.49}$ (n=14)	lDDT$_{\text{PLI}}$ (n=12)	RMSD$_{<1}$ (n=12)	RMSD$_{<2}$ (n=12)
Small								
Transformer	68	86	0.35	0.22	0.17	0.21	0.06	0.10
Pairformer (Boltz-1)	68	125	0.39	0.46	0.24	0.36	0.10	0.21
Pairmixer (Ours)	68	98	0.37	0.39	0.21	0.35	0.06	0.16
Medium								
Transformer	68	128	0.35	0.19	0.16	0.27	0.04	0.15
Pairformer (Boltz-1)	68	194	0.38	0.66	0.35	0.39	0.14	0.23
Pairmixer (Ours)	68	146	0.39	0.49	0.39	0.38	0.12	0.24
Large								
Transformer	68	173	0.36	0.29	0.16	0.26	0.06	0.10
Pairformer (Boltz-1)	68	290	0.41	0.68	0.43	0.37	0.12	0.31
Pairmixer (Ours)	68	192	0.38	0.50	0.35	0.34	0.12	0.23
Large Phase 2								
Transformer	20	232	0.37	0.34	0.17	0.26	0.11	0.11
Pairformer (Boltz-1)	20	421	0.42	0.64	0.43	0.36	0.10	0.28
Pairmixer (Ours)	20	269	0.41	0.52	0.36	0.34	0.14	0.31
Boltz-1 public model								
Pairformer (Boltz-1)	-	-	0.4	0.68	0.43	0.45	0.23	0.42

C.6 FULL BIOMOLECULAR STRUCTURE PREDICTION RESULTS

Table 4 and Table 5 report the full set of evaluation metrics across all architectures, along with the number of complexes evaluated by each metric. We retrain Boltz-1 for our Pairformer baselines and additionally include comparisons against the public checkpoint.

D TRAINING HYPERPARAMETERS

Table 6 includes a thorough list of the hyperparameters used for our experiments. This table additionally includes the training FLOPs for all model architectures and sizes.

Table 6: **Structure Prediction Hyperparameters.** Dashes (-) indicate that the value is the same as the previous column. The large model uses a multi-stage training approach: first with smaller crops and mixed data, second with larger crops and PDB-only data.

Hyperparameter	Small	Medium	Large Stage 1	Large Stage 2
Model Architecture				
Number of Backbone Layers	12	24	48	48
Number of MSA Layers	4	-	-	-
Token representation dim (C_s)	384	-	-	-
Pair representation dim (C_z)	128	-	-	-
Backbone dropout	0.25	-	-	-
MSA Module dropout	0.15	-	-	-
Number of Diffusion Layers	6	24	24	24
Atom representation dim	128	-	-	-
Atom pair representation dim	16	-	-	-
Training				
Optimizer	Adam	-	-	-
Maximum learning rate	1.8×10^{-3}	-	-	-
Diffusion multiplicity	16	-	-	-
Recycling	0-3	-	-	-
Epochs	68	68	68	20
Training Samples	6.8M	6.8M	6.8M	2M
Data Processing				
Data source	PDB + OpenFold	-	-	PDB
Maximum tokens	384	384	384	512
Maximum atoms	3,456	3,456	3,456	4,608
Maximum MSA sequences	2,048	-	-	-
Samples per epoch	100,000	-	-	-
Total Batch size	128	-	-	-
Inference				
Number of sampling steps	200	-	-	-
Maximum MSA Sequences	4096	-	-	-
Recycling	10	-	-	-
Diffusion samples	5	-	-	-
Training Infrastructure				
GPU Type	H200	-	-	-
Number of GPUs	32	32	32	64
Total Training FLOPs				
Boltz-1 (Pairformer)	8.306e+19	1.467e+20	2.707e+20	1.572e+20
Pairmixer	4.817e+19	8.557e+19	1.572e+20	8.716e+19
Transformer	5.784e+18	7.888e+18	8.941e+18	4.205e+18
Inference FLOPs				
Boltz1 (Pairformer)	9.100e+15	1.595e+16	2.964e+16	-
Pairmixer	4.474e+15	7.849e+15	1.460e+16	-
Transformer	4.137e+14	4.975e+14	6.652e+14	-