STATE-SPACE, ATTENTION OR HYBRID? PRIMA: A FAMILY OF FOUNDATION MODELS FOR PROTEIN-PROTEIN INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein–protein interactions (PPIs) govern essential biological processes, yet the vast majority of protein language models (PLMs) are transformers trained on single sequences. Our goal is to encode the semantics of the interaction: we ask which architectures are best suited to model PPIs, whether the sequence mixer matters, and at what scale it begins to. We present PRIMA, a family of PPI foundation models that share one training recipe and differ only in their sequence mixer: bidirectional state-space layers (PRIMA-M), full FlashAttention (PRIMA-A), and a hybrid that interleaves one attention layer every k state-space layers (PRIMA-H). Each is trained at 8M and 150M parameters in two phases, pre-training on single sequences and post-training on concatenated PPI pairs. We evaluate against the ESM2 family and MINT, the state-of-the-art PPI specialist at 813M parameters, and we rebuild SKEMPI into SKEMPI-R to expose homology leakage and models that resort to recall. At 8M the benchmarks cannot separate the mixers. At 150M they separate, and PRIMA-H matches or outperforms models several times larger. On SKEMPI-R it is first on every metric, reaching Pearson 0.451 against 0.417 for ESM2 650M ($4.3\times$ larger) and 0.401 for MINT ($5.4\times$ larger), with a partition-to-partition spread $6.7\times$ tighter, and it leads on the rows where models cannot exploit recall. On Gold Standard PPI, the classification benchmark built to remove protein-identity shortcuts, it matches MINT and PRIMA-M. Scaling PRIMA-H to 650M does not improve its ability to encode interaction specific properties, it only buys a better language model: the extra capacity goes into representing each chain rather than the relationship between them, a different approach is necessary. PRIMA mixers are equivalent after pre-training and diverge after post-training, placing attention's deficit in the interaction objective rather than in protein modelling. We find that PRIMA-H needs no positional encoding at all: permutation-equivariant attention layer can use the position encoded by state-space layers, previously demonstrated only for causal models. At 16,384 tokens PRIMA-H 150M encodes a complex in 74 ms and 0.8 GB against 10.5 s and 78 GB for MINT, and runs 65,536-token inputs on which MINT runs out of memory on an 80 GB GPU; below $\sim$2,000 tokens the state-space stacks are launch-bound and hold no speed advantage. PRIMA shows that a mixed state-space/attention stack produces specialist-level PPI representations at a fraction of the size and training budget.

1 INTRODUCTION

Protein-protein interactions (PPIs) regulate essential biological processes, from immune response and metabolic control, to signal transduction and gene regulation; thus, their study is crucial for understanding cellular mechanisms, identifying disease pathways, and guiding drug discovery (Greenblatt et al., 2024; Nada et al., 2024). Having a deep learning system able to model PPIs efficiently is an actively researched topic (Lee, 2023) that presents two fundamental problems: a lack of large, well curated PPI datasets (Kovtun et al., 2024); the computational cost that comes with the size of interacting protein complexes (Celik & Xie, 2024). Whereas a large portion of the field leverages structure data (Kiouri et al., 2025), we direct our attention to protein language models (PLMs): ex-

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.

perimentally acquired sequences are far more abundant and less noisy than structures, and by solving a masked language modeling (MLM) task on large corpora a model implicitly learns evolutionary, structural, and functional constraints (Elnaggar et al., 2021; Lin et al., 2022). The vast majority of state-of-the-art PLMs are built on self-attention (Nijkamp et al., 2022; Elnaggar et al., 2021; Lin et al., 2022), whose computational cost scales quadratically with sequence length ($O(L^2)$). Given that multi-chain complexes often span thousands of amino acids, scalable foundation models for PPIs remain scarce, and the space of sequence mixers for PPI modelling is largely unexplored. The strongest PPI specialist, MINT (Ullanat et al., 2026), encodes each chain with an ESM2 650M backbone and couples them through cross-attention, and caps training at 512 tokens per chain. An initial attempt at efficient PPI modelling without self-attention is the proof-of-concept Fiorellini-Bernardis et al. (2026): an $\sim$8M-parameter pure bidirectional Mamba1 model (Gu & Dao, 2024), trained on subsets of UniRef50 and STRING and compared against a size-matched encoder-only transformer.

Our goal is to find the best way of encoding the semantics of the interaction between chains in protein complexes. We test whether solving MLM on concatenated PPI sequences allows the model to learn the joint statistics of the complex and interaction-relevant patterns without experimental labels. To do so, we explore state-of-the-art architectures, we deploy them to model long input PPIs and assess how they solve the task at different scales. We introduce PRIMA, a family of PPI foundation models that share a single training recipe and differ only in their sequence mixer: PRIMA-M, a stack of bidirectional state-space layers built on Mamba2 (Dao & Gu, 2024); PRIMA-A, a stack of full FlashAttention layers (Dao et al., 2022); PRIMA-H, a hybrid stack that interleaves one attention layer every k state-space layers (Waleffe et al., 2024). We train every variant at 8M and 150M parameters, and we scale PRIMA-H, the best performing variant, up to 650M. We pre-train PRIMA MLM on single sequences to learn general protein representations, then continued MLM on concatenated PPI pairs for interaction-specific context.

We test PRIMA against the best PPI specialist in the literature MINT (Ullanat et al. (2026), 813M parameters) and the ESM2 family (arguably the most widely used PLM), to control whether a good single sequence PLM can solve the task and if MLM is effective on PPIs. Because of the scarcity and weakness of existing PPI benchmarks, we design SKEMPI-R, a dedicated binding affinity benchmark that exposes homology leakage and recall exploitation, and use it to evaluate all the models on SKEMPI-R; we also test on three binding prediction benchmarks that the MINT authors also use.

Our contributions are:

- **A controlled comparison of sequence mixers for multi-chain protein modelling.** Three mixers, identical data, recipe and parameter count, at multiple scales, with cost measured.
- **SKEMPI-R.** A rebuilt binding-affinity regression benchmark that exposes homology leakage and tests when the models exploit recall to solve it.
- **A scale-dependent answer.** At 8M the mixer is irrelevant. At 150M PRIMA-H leads on every metric. At 650M you get a better language model, not a better interaction model.
- **Specialist-level PPI representation at a fraction of the size.** PRIMA-H 150M ranks first or tied-first across the PPI benchmarks, ahead of ESM2 650M and MINT at 813M.
- **A sequential state-space and attention stack does not need positional encodings.** Permutation-equivariant attention layers use the sequentiality encoded by the state-space layers.

2 Related work

Few protein language models tackle protein interactions. MINT (Ullanat et al., 2026) is the closest to this work: at 813M parameters it encodes each chain with an ESM2 650M backbone and couples them through cross-attention blocks, trained with MLM on a curated STRING split of physically interacting pairs; the quadratic cost of attention forces a 512-token cap per chain during training, with longer chains randomly cropped. LC-PLM (Wang et al., 2024) is a 1B-parameter stack of bidirectional Mamba1 layers, similar to PRIMA-M, trained on UniRef50 and extended as LC-PLM-G to multiple proteins through random walks on an interaction graph. It is never trained on concatenated complexes and never evaluated on binding affinity. ProtMamba (Sgarbossa et al., 2024) is a 107M causal Mamba trained with a fill-in-the-middle objective on sequences concatenated from the same multiple sequence alignment; it targets within-family tasks such as mutational effect prediction, not interaction between different proteins.

3 METHODS

3.1 ARCHITECTURE

We build three variants at 8M and 150M: PRIMA-M as a stack of state-space layers, PRIMA-A as a stack of self-attention layers, PRIMA-H as a hybrid stack of sequential state-space and self-attention layers. The three variants share everything except the sequence mixer: tokenizer, block wiring, tied LM head, and training recipe. To keep the parameter counts matched we adjust depth and the gated-MLP ratio, never the model width (Table 1). An attention mixer is lighter than a Mamba2 mixer at equal width, so PRIMA-A needs a wider MLP to land at the same size.

- **PRIMA-M**: a stack of N bidirectional Mamba2 blocks. We use bidirectional blocks because a causal model would give an asymmetric view of a complex: with the chains concatenated, no position in chain A could ever see chain B, while every position in B sees all of A. Running both directions doubles the sequence-mixing compute but leaves the $O(L)$ scaling intact.
- **PRIMA-A**: the same blocks with every mixer replaced by full non-causal self-attention. With no state-space layer left the model has no notion of order, so rotary positional encodings are required here.
- **PRIMA-H**: the PRIMA-M stack with one in every k blocks replaced by an attention block, k chosen so that attention accounts for 15–20% of the mixers, as in NVIDIA (2025). The first and last blocks are always state-space, so the representation read by the LM head and by every probe comes from an SSM layer. The attention layers carry no positional encoding at all, instead provided by the state-space layers. This is common practice for causal hybrid models (Lieber et al., 2024), but there attention can recover order from the mask alone (Haviv et al., 2022). PRIMA-H is bidirectional, so its attention layers are permutation-equivariant and the state-space layers are their only possible source of order.

Single sequences are tokenised as `<cls>` A and pairs as `<cls>` A `<inter><cls>` B, over a 30-token vocabulary (20 standard amino acids, 5 non-standard or unknown residues, and 5 special tokens). Appendix B.1 gives the remaining implementation details.

Table 1: The PRIMA family. All variants share the tokenizer, the tied LM head, and the block wiring; the gated-MLP ratio and depth are adjusted so the parameter counts match, since an attention mixer is lighter than a Mamba2 mixer at equal width.

scale	variant	d_{model}	layers	mixer layout	MLP ratio	params
8M	PRIMA-M	256	9	9 SSM	2.0	7.8M
	PRIMA-H	256	10	8 SSM + 2 attn	2.0	8.2M
	PRIMA-A	256	9	9 attn	3.0	7.7M
150M	PRIMA-M	640	30	30 SSM	2.0	153.2M
	PRIMA-H	640	31	26 SSM + 5 attn	2.0	153.2M
	PRIMA-A	640	30	30 attn	2.8	152.5M

3.2 TRAINING STRATEGY

We pre-train on UniRef50 (The UniProt Consortium, 2023), sampling a different UniRef90 member of each cluster at every epoch as ESM2 does: this averages 1.9 sequences per cluster. We post-train on the STRING v12 (Szklarczyk et al., 2023) split curated by MINT: 96M physically interacting pairs for training and 250k held out. Sequences are never cropped, in either phase.

Every variant is trained in two phases with the same objective: masked language modelling on 15% of tokens, following the BERT protocol. Pre-training runs for 10 epochs on UniRef50, about 190B tokens. Post-training continues for 2 epochs on the 96M physically interacting STRING pairs curated by MINT, about 165B tokens, with all parameters unfrozen. Batches are built to a fixed token budget rather than a fixed number of sequences, so padding stays low across a wide length distribution. The learning rate follows a warm-up–stable–decay schedule. We search the peak rate separately for each model and keep the values as close as possible within a scale, so that no variant is handicapped by a schedule tuned for another. At 150M this leaves one difference: PRIMA-A trains best at 7×10^{-4} against 9×10^{-4} for PRIMA-M and PRIMA-H. Everything else is identical.

By the end of pre-training the 8M variants are capacity-saturated and had stopped improving for three epochs, while the 150M variants were still improving and would benefit from a longer run; we see no overfitting at either scale. Appendix B.2 gives the full schedule, the batching parameters and the per-model values.

3.3 SKEMPI-R: REBUILDING THE BINDING-AFFINITY BENCHMARK

SKEMPI 2.0 (Jankauskaitė et al., 2019) holds 6,706 point mutations across 345 complexes with experimentally measured $\Delta\Delta G$. The authors of MINT evaluated their model following conventional protocol which assigns complexes to three folds at random and averages the correlation obtained on each (Jin et al., 2023). We find four fundamental problems with it, and we address each with a fix: our goal is to build a benchmark that quantifies how well the notion of interaction is encoded in each model's representation. Appendix C shares a detailed analysis.

- **The folds are drawn at random, and unseeded.** MINT's folds cannot be regenerated nor the results reproduced. Moreover, a correlation is bounded by the spread of the target it is scored against, and a test fold that happens to collect $\Delta\Delta G$ complexes with a spread much narrower than what seen in training depresses every model that is tested on it. We find that a draw is damaged 45% of the time. To address this, we match three totals per fold, namely the number of rows, $\sum \Delta\Delta G$ and $\sum \Delta\Delta G^2$. This aligns each fold's mean and spread to its own training set's, and we repeat the whole cross-validation over five seeded partitions. Our folds reach a spread ratio of 0.988–1.011 and a mean gap of at most 0.003 kcal/mol.
- **Holding out complexes does not hold out proteins.** Different complexes share homologous chains, so 72.9% of test rows have a chain at least 90% identical to one in training. A lookup table that predicts a row from the mean $\Delta\Delta G$ of the same mutation seen in a different complex can answer up to 46% of a fold, and on those rows scores Pearson 0.479. In other words, a model with no knowledge of the protein interface (e.g. ESM2) can score high simply by exploiting recall. To address this, we split the test rows by whether the same mutation appears on a homologous chain in training, and report every metric on both splits. A margin that survives on the rows without such a counterpart is about representation rather than recall.
- **Averaging per-fold correlations rewards inconsistency.** A correlation is invariant to rescaling the prediction, so scoring inside each fold hands the model a separate intercept and slope in every fold. Once a model is deployed it has only one. Measured on the same predictions and folds, averaging rather than pooling inflates every model, by 0.03 to 0.09 Pearson, and reorders them. To address this, we pool the out-of-fold predictions and score once.
- **A correlation does not say whether a prediction is useful**, and 62% of the $\Delta\Delta G$ variance in SKEMPI lies within complexes rather than between them, which is not what a pooled correlation measures. We report four quantities: pooled Pearson and Spearman, the mean Spearman within each complex, and the percentage by which a model's RMSE improves on a predictor that always returns the training mean.

We probe the frozen final-layer embeddings with ridge regression rather than an MLP, for three reasons. Ridge has a closed-form solution, so refitting returns the same answer and the spread we report comes from the folds alone. It has one scalar penalty, it does not hand a wider model a larger probe. With roughly 4,500 training rows per fold against embeddings of 1,280 to 2,560 dimensions, an MLP of the size MINT uses would carry more parameters than there is data to fit them. The cost is that ridge reads only what is linearly decodable. Features are extracted as difference between the wild-type and mutant embeddings to align all three model families (more in Appendix C). We also evaluate under MINT's original SKEMPI protocol as a check (see Appendix D)

3.4 PPI CLASSIFICATION BENCHMARKS

We evaluate on three binary interaction-classification benchmarks, using MINT's splits and protocol. Gold Standard PPI (Bernett et al., 2024) is scored as AUPRC on 52,048 class-balanced test pairs; Human PPI and Yeast PPI (Xu et al., 2022) as accuracy on 180 and 394 test pairs. Only Gold Standard measures interaction: Human PPI and Yeast PPI are protein-disjoint re-splits of the PAN (Pan et al., 2010) and GUO (Guo et al., 2008) datasets, on which Bernett et al. (2024) showed that negatives are sampled such that a protein appears almost entirely in positive or almost entirely in

negative pairs. Re-splitting by protein removes the overlap but not that shortcut. Appendix E shows that a linear probe on frozen ESM2 650M embeddings with no knowledge of the interaction reaches 0.82 AUROC on Human PPI and 0.72 on Yeast PPI. On Gold Standard PPI it is instead at chance, 0.52. We therefore rank models on Gold Standard PPI, and report the other two for continuity with prior work.

The features are extracted from the two chains concatenated, so an additive model has no interaction term and a linear probe is not an option as it is for SKEMPI-R. We use MINT's as reported in the paper: one hidden layer of width 640, we select on the validation split, we report the mean and standard deviation over three probe-training repetitions. A disclaimer: at the moment of the writing, MINT's published code currently shows a much bigger hidden layer of size 2560. Appendix E gives the full split sizes, the per-model readouts and the cropping.

3.5 Efficiency and scaling benchmark

We measure the latency and peak memory of a single forward pass over input lengths from 128 to 65,536 tokens on one NVIDIA H100 80GB, at batch sizes 1 and 8. Every model runs in bf16 with the fastest attention kernel available to it: FlashAttention for PRIMA-A and PRIMA-H, PyTorch's block-wise `scaled_dot_product_attention` for ESM2, and its own implementation for MINT. A block-wise kernel never materialises the $L \times L$ score matrix, so ESM2's memory grows linearly with length and only its arithmetic stays quadratic; benchmarking attention against a naive implementation would overstate the comparison. MINT consumes a chain pair, so an input of length L reaches it as two chains of $L/2$. Appendix G gives the timing protocol and further details.

4 Results

4.1 Held-out single-sequence modelling

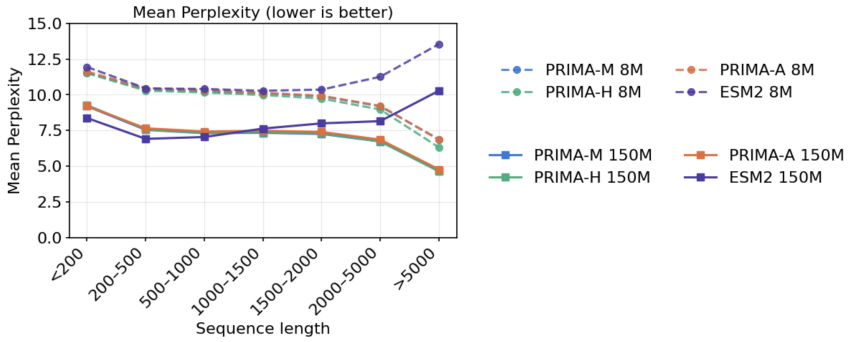


Figure 1: Mean perplexity of PRIMA and ESM2 on a held-out test set of single sequences from STRING, binned by input length. PRIMA models are tested after the pre-training phase, ∼190B tokens on UniRef50 against the ∼1T tokens of ESM2.

After pre-training and before any PPI-specific training, we measure how well each model predicts masked residues in 181,883 held-out single chains drawn from STRING that do not appear in the UniRef50 corpus, with no length cap (A detailed description of the test is given in Appendix F). Below 1000 residues, at 8M ESM2 and PRIMA variants are level, at 150M ESM2 is ahead of every PRIMA variant, consistent with the fact that PRIMA trains on ∼190B tokens, and ESM ∼1T. Since 82.6% of the set is shorter than that, ESM2 is ahead on the pooled average too. Between 1000 and 1500 the ordering reverses, and from there the gap widens monotonically, above 5000 residues ESM2 150M sits at 10.28 against PRIMA-M 150M's 4.66 (Figure 1). The crossover is at ESM2's 1024-token training cap, and past it the two families move in opposite directions: PRIMA improves with length, as longer chains carry more context, while ESM2 degrades. This is due to the training length distribution cap, not the mixer, and PRIMA-A is the evidence: it is a pure attention stack with RoPE and it tracks the two state-space variants in every bin because it was trained without a cap. The interesting finding is that the three PRIMA variants stay within 0.016 nats of each other at both scales, so PRIMA-H pays nothing for carrying no positional encoding at all: as

mentioned in Section 3.1, its attention layers are bidirectional and therefore permutation-equivariant, and sequence order reaches them only through the state-space layers. That this works had been demonstrated only in causal models (Lieber et al., 2024; Haviv et al., 2022). The three PRIMA variants differ only slightly, with PRIMA-A lagging behind, and can be considered equivalent. A detailed breakdown of the results is given in Appendix F.

4.2 Results on SKEMPI-R

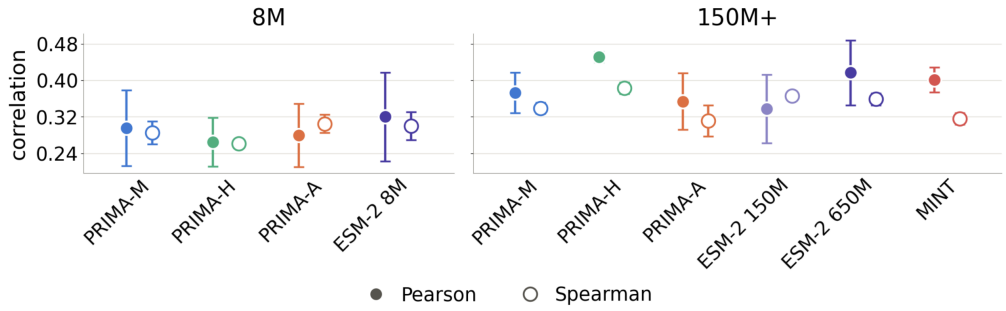


Figure 2: Performance of PRIMA, ESM2 and MINT on pooled out-of-fold predictions on SKEMPI-R. Mean and standard deviation are calculated over the 5 seeded split partition. (a) reports Pearson and Spearman correlation coefficient mean and standard deviation for PRIMA 8M and ESM2 8M; (b) reports Pearson and Spearman correlation coefficient mean and standard deviation for PRIMA 150M, ESM2 150M / 650M and MINT.

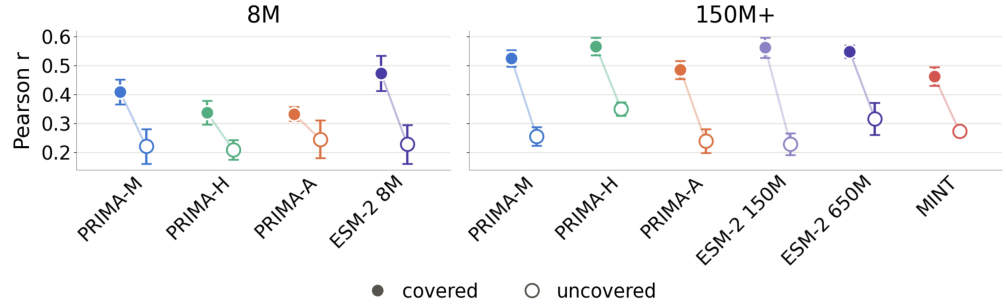


Figure 3: Pearson r of PRIMA, ESM2 and MINT on covered (mutation seen in a homologous chain in a different complex during training) versus uncovered (mutation never seen) rows. Mean and standard deviation are calculated over the 5 seeded split partition. (a) reports the correlation coefficient mean and standard deviation for PRIMA 8M and ESM2 8M; (b) reports the correlation coefficient mean and standard deviation for PRIMA 150M, ESM2 150M / 650M and MINT.

PRIMA-H 150M leads the board. As shown in Figure 2 (b), PRIMA-H 150M is first on every metric, and is the most stable model across partitions. The hybrid reaches Pearson 0.451 and Spearman 0.383 against 0.417 and 0.359 for ESM2 650M ($4.3\times$ larger) and 0.401 and 0.316 for MINT ($5.4\times$ larger). PRIMA-H 150M also leads on covered (where the model is tested on the exact mutation seen in a homologous chain during training, although appearing in a different complex) and uncovered mutations (with the model being tested on unseen mutations), with the smallest gap between the two (Figure 3 (b)): the hybrid's embeddings encode useful information about the interaction and rely on recall the least. This analysis exposes where the performance of ESM2 comes from: it relies on rows that can be solved without knowledge of the interaction. The margin over a do-nothing predictor and the ranking of mutations within each complex (per-complex Spearman) are also reported in Table 2; both are led by PRIMA-H. The spread matters: PRIMA-H's partition-to-partition standard deviation is 0.011 on Pearson against ESM2 650M's 0.071, a factor of 6.7, and 0.41 against 5.76 on the margin over the null, a factor of 14. Since the five partitions are moment-matched by construction, a large spread across them means a model's score depends on which complexes it's tested on. Also, half of ESM2's feature vector is structurally zero on the

rows where only one side is mutated: it encodes the two chains in separate forward passes, so the non-mutated side's embedding is bit-identical between wild-type and mutant. ESM2 is scoring on the mutated chain alone, and still scores well on pooled Spearman: this demonstrates that SKEMPI can be partly solved without using the interaction.

Scale does not continue the trend. We trained the hybrid at 650M to test whether the improvement from 8M to 150M continues. It does not. PRIMA-H 650M is a clearly better language model, held-out cross-entropy 1.695 against 1.985, and appears worse here, at Pearson 0.417 against 0.451. However, probing each layer of the models (Figure 7, Appendix A) shows that the best embeddings for the task are not at the last layer, and the two models are equivalent. The 650M model is ahead on the mutated chain at almost every depth and behind on its partner, meaning the extra capacity went into representing the chains rather than the relationship between them. Masked language modelling on PPIs returns a better language model, not a better interaction model. Indeed, the performance of PRIMA-H 650M is similar to that of ESM2 650M. Appendix A gives the full analysis.

The result does not depend on our benchmark. A fair objection to SKEMPI-R would be that we designed it and then won on it. We therefore evaluated every model a second time under MINT's protocol (complex-disjoint folds drawn at random, MLP probe, average per-fold correlations) and PRIMA-M and PRIMA-H 150M are still ahead of every baseline (Figure 8, Appendix D). The redesign makes the margin trustworthy and tighter.

At 8M the task is not solved. No model at 8M improves on the do-nothing predictor by more than 2.4% (Table 2), against partition-to-partition spreads of 4.9 to 10.3 percentage points, so every one of those margins is consistent with zero. The metrics also point in different directions: ESM2 8M is first on Pearson but worst on RMSE vs. null, PRIMA-A is first on both pooled and per-complex Spearman, and PRIMA-H, first on everything at 150M, is last on Pearson. We read this as the benchmark having no resolving power at 8M rather than as evidence about the mixers.

Per-complex ranking is hardly solved. Mean per-complex Spearman ranks the mutations within one complex, which is the question an experimentalist might ask: all models sit between 0.06 and 0.15. PRIMA-H leads it, but at 0.152 the within-complex ranking problem is essentially open. Since 62% of the $\Delta\Delta G$ variance in SKEMPI is within complex, the pooled correlations that the literature reports are largely measuring the easier between-complex problem.

RMSE vs. null exposes useless predictions. ESM2 150M reaches Pearson 0.338 and the second-best Spearman in the table, 0.366, yet its RMSE is on average 3.7% worse than simply returning the training mean, with a spread of 17 percentage points. Correlation is invariant to affine rescaling of the prediction, so a model can rank mutations reasonably while placing them on the wrong scale. This is why we report the margin over the null alongside the correlations, and it calls for caution for any SKEMPI result reporting only correlation.

4.3 Results on the PPI classification benchmarks

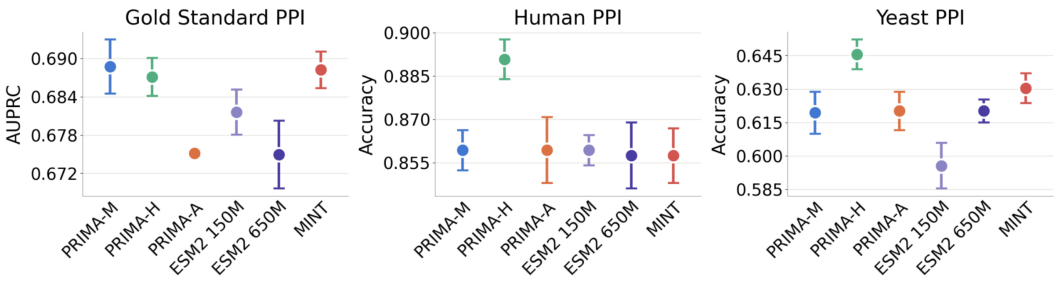


Figure 4: Results of PRIMA-M 150M, PRIMA-H 150M, PRIMA-A 150M, ESM2 150M, ESM2 650M and MINT on Gold Standard PPI, Human PPI and Yeast PPI. Points are the mean and standard deviation over 3 probe-training repetitions.

Table 2: SKEMPI-R results. Values are mean ± standard deviation over five seeded partitions, with out-of-fold predictions pooled within each partition before scoring. We report Pearson, Spearman, per-complex ρ (mean Spearman within a complex); percentage reduction in RMSE (kcal/mol) relative to a predictor that always returns the training mean. PRIMA-H 650M is reported as scaling ablation, see Appendix A

model	Pearson	Spearman	per-complex Spearman	RMSE vs. null (%)
PRIMA-M 8M	0.296 ± 0.083	0.285 ± 0.025	0.058 ± 0.006	2.17 ± 7.56
PRIMA-H 8M	0.265 ± 0.053	0.262 ± 0.005	0.065 ± 0.008	1.67 ± 5.57
PRIMA-A 8M	0.280 ± 0.069	$\mathbf{0.305} \pm 0.020$	$\mathbf{0.096} \pm 0.008$	$\mathbf{2.39} \pm 4.87$
ESM2 8M	$\mathbf{0.320} \pm 0.097$	0.300 ± 0.031	0.063 ± 0.023	0.94 ± 10.26
PRIMA-M 150M	0.373 ± 0.044	0.339 ± 0.012	0.112 ± 0.023	5.40 ± 4.42
PRIMA-H 150M	$\mathbf{0.451} \pm 0.011$	$\mathbf{0.383} \pm 0.012$	$\mathbf{0.152} \pm 0.006$	$\mathbf{9.83} \pm 0.41$
PRIMA-A 150M	0.354 ± 0.062	0.311 ± 0.034	0.093 ± 0.021	4.65 ± 3.90
ESM2 150M	0.338 ± 0.075	0.366 ± 0.007	0.113 ± 0.014	-3.73 ± 16.95
ESM2 650M	0.417 ± 0.071	0.359 ± 0.013	0.137 ± 0.019	7.89 ± 5.76
MINT 813M	0.401 ± 0.027	0.316 ± 0.012	0.108 ± 0.019	7.73 ± 1.79
PRIMA-H 650M	0.417 ± 0.032	0.378 ± 0.029	0.121 ± 0.011	7.96 ± 1.93

At 8M no consistent signal emerges. The three benchmarks pick three different winners: PRIMA-M and PRIMA-H tie on Gold Standard PPI (AUPRC 0.67), ESM2 8M leads Human PPI with all four models overlapping, and PRIMA-A leads Yeast PPI (Figure 9, Appendix D). According to the results on SKEMPI-R (Section 4.2) at this scale, we read the 8M results as the evaluation having no resolving power.

At 150M PRIMA-H is first. On Gold Standard PPI the top of the field is a three-way tie (Figure 4): PRIMA-M 0.689, MINT 0.688 and PRIMA-H 0.687 are mutually indistinguishable, and all three are above ESM2 and PRIMA-A. On Human PPI, PRIMA-H reaches 0.887 above every other model (between 0.858 and 0.860) and with no overlap. On Yeast PPI, PRIMA-H reaches 0.645 against MINT's 0.630, again without overlap, with ESM2 650M, PRIMA-A and PRIMA-M level at 0.619–0.620 and ESM2 150M last at 0.596. PRIMA-H is therefore tied first on Gold Standard PPI, and first on Human and Yeast, which we report for continuity but do not rank on. PRIMA-H 150M confirms what SKEMPI-R showed in Section 4.2.

PRIMA-A is weakest at both scales. Consistent with Section 4.2, pure attention is the weakest overall PRIMA variant on Gold Standard PPI at 8M and at 150M, in both cases with non-overlapping intervals, and Gold Standard PPI is the benchmark of the three built specifically to remove the shortcuts that let a model score without modelling interaction.

Neither family gains from scale on these benchmarks. ESM2 650M is behind ESM2 150M on Gold Standard PPI (0.675 against 0.682), level on Human PPI, and ahead on Yeast PPI. The MINT paper reports the same pattern comparing ESM2 650M with ESM2 3B. PRIMA-H behaves no differently: at 650M it gains 0.006 AUPRC on Gold Standard PPI, while losing on Human and Yeast PPI (Appendix A). These benchmarks do not seem to reward capacity.

4.4 Efficiency and scaling

Figure 5 shows the inference time of all the models tested in the work at batch size 1 and 8. Table 7 in Appendix G reports the detailed latency and peak memory for all the models tested in this work at batch size 8. The test is conducted on a single NVIDIA H100 80GB GPU. Latency is reported in milliseconds, memory in GB, as a function of sequence lengths L. The only model ever exhausting the H100's memory is MINT: the largest single input that MINT can process before OOM is a single sequence of $L = 16{,}384$: this gets encoded in 10.5 s with 78.0 GB, against 73.8 ms and 0.80 GB for PRIMA-H 150M.

Table 8 in Appendix G shows the scaling exponent exhibited by each model in computation cost and memory. Fitting is expressed as cost $\propto (\text{tokens})^k$: as expected, the arithmetic cost of PRIMA-M grows linearly $k \simeq 1.00$; PRIMA-A reaches 1.79, short of 2.0 because its wider gated MLP keeps the linear term relevant further out; PRIMA-H sits in between. ESM2 shows the expected quadratic behaviour. The mamba-ssm implementation is slowest at shorter inputs due to kernel

launch: PRIMA-M 150M is $4.6\times$ slower than ESM2 150M at 512 tokens and overtakes it only from 2,048. All models except MINT grow linearly in memory ($k = 0.91$–0.98), and at equal parameter count PRIMA has no advantage: PRIMA-M 150M peaks at 15.0 GB against ESM2 150M's 14.3 GB. MINT is the exception at $L^{1.63}$: MINT is measured as published, so its memory scaling and boundaries are a property of its implementation. Part of PRIMA-A's margin over ESM2 150M at long inputs is a kernel effect rather than an architectural one: ESM2 150M uses head dimension 32 against PRIMA-A's 64, and block-wise attention kernels are inefficient at the smaller value.

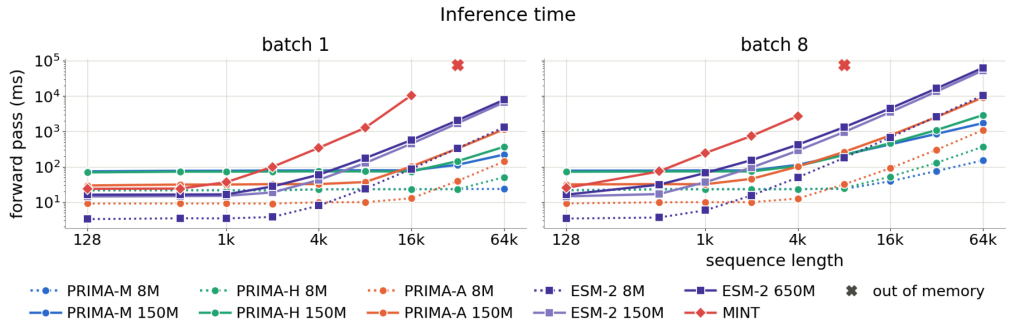


Figure 5: Inference time of all the models tested in the work at batch size 1 and 8. The time is reported in milliseconds as a function of the individual sequences length.

5 Conclusions

We set out to ask what the best way is to represent protein-protein interactions with a language model, whether the sequence mixer matters and at what scale. We built three models that differ only in their mixer, bidirectional state-space (PRIMA-M), full attention (PRIMA-A), and a hybrid stack (PRIMA-H); we trained all of them at 8M and 150M parameters, the hybrid also at 650M, in two distinct phases with an identical recipe. We found that the mixer does matter: it matters significantly more for PPIs than for single proteins, and that at 8M none of the benchmarks can see it.

PRIMA-H at 150M is the strongest model we tested. It is first on every metric of SKEMPI-R and the most stable across re-draws of the split; it shares first place on Gold Standard PPI, built to resist shortcuts, and it does so at $5.4\times$ fewer parameters than MINT and significantly less training budget. Where PRIMA variants separate is informative in itself: on single sequences they are almost equal and they diverge only after post-training on concatenated pairs, which places pure attention's deficit in the multi-chain objective. Cost follows the architecture as designed: PRIMA-M and PRIMA-H are less efficient than PRIMA-A for short inputs and small batch sizes, increasingly more efficient for long inputs. We also find that PRIMA-H does not need to carry positional encoding of any kind. Permutation-equivariant attention layers can exploit sequentiality encoded by the state-space layers before them.

What none of these models fully solves is what an experimentalist asks: to rank mutations within individual complexes. Spearman sits between 0.06 and 0.15 for every model tested, with PRIMA-H 150M on top. Since 62% of the $\Delta\Delta G$ variance in SKEMPI is within complex, that is the part of the problem the field's pooled correlations have been least informative about.

The question of encoding the semantics of the interaction remains, at least partly, open. Scaling PRIMA to even larger parameter counts buys a better language model but not a better interaction model. Solving masked language modelling on concatenated protein sequences means collapsing a complex high-dimensional embedding manifold to predict one of twenty amino-acids. Since masking positions directly involved in the interface is a rare event, and residue identity is mostly determined by the local environment, MLM on PPI can be solved without knowledge of the interaction. Extra parameter capacity makes it even easier for the model to ignore the information from partners across the interface. That single-sequence models with no PPI training remain competitive on these benchmarks is consistent with this. We show that MLM encodes some knowledge of the interaction, but we believe a different training task that forces the models to exploit partner information is necessary to truly model PPIs.

REFERENCES

Judith Bernett, David B Blumenthal, and Markus List. Cracking the black box of deep sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, 25(2):bbae076, 2024. doi: 10.1093/bib/bbae076.

Muhammed Hasan Celik and Xiaohui Xie. Efficient inference, training, and fine-tuning of protein language models. *bioRxiv*, 2024. doi: 10.1101/2024.10.22.619563. URL https://www.biorxiv.org/content/early/2024/10/25/2024.10.22.619563.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Wang Yu, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3095381.

Arturo Fiorellini-Bernardis, Sebastien Boyer, Jean Quentin, Ghassene Jebali, and Oliver Bent. PRIMA: a bidirectional state-space architecture and training approach for sequence modelling of protein-protein interactions. In *ICML 2026 3rd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences*, 2026. URL https://openreview.net/forum?id=9OVKsH5DXt.

Jack F. Greenblatt, Bruce M. Alberts, and Nevan J. Krogan. Discovery and significance of protein-protein interactions in health and disease. *Cell*, 187(23):6501–6517, 2024. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2024.10.038. URL https://www.sciencedirect.com/science/article/pii/S0092867424012534.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. 2024. URL https://arxiv.org/abs/2312.00752.

Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030, 2008. doi: 10.1093/nar/gkn159.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information, 2022. URL https://arxiv.org/abs/2203.16634.

Justina Jankauskaitė, Brian Jiménez-García, Justas Dapkūnas, Juan Fernández-Recio, and Iain H Moal. SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019. doi: 10.1093/bioinformatics/bty635.

Wengong Jin, Siranush Sarkizova, Xun Chen, Nir Hacohen, and Caroline Uhler. Unsupervised protein-ligand binding energy prediction via neural euler's rotation equation, 2023. URL https://arxiv.org/abs/2301.10814.

Despoina P. Kiouri, Georgios C. Batsis, and Christos T. Chasapis. Structure-based approaches for protein–protein interaction prediction using machine learning and deep learning. *Biomolecules*, 15(1), 2025. ISSN 2218-273X. doi: 10.3390/biom15010141. URL https://www.mdpi.com/2218-273X/15/1/141.

Daniel Kovtun, Mehmet Akdel, Alexander Goncearenco, Guoqing Zhou, Graham Holt, David Baugher, Dejun Lin, Yusuf Adeshina, Thomas Castiglione, Xiaoyun Wang, Céline Marquet, Matt McPartlon, Tomas Geffner, Gabriele Corso, Hannes Stärk, Zachary Carpenter, Emine Kucukbenli, Michael Bronstein, and Luca Naef. Pinder: The protein interaction dataset and evaluation resource. *bioRxiv*, 2024. doi: 10.1101/2024.07.17.603980. URL https://www.biorxiv.org/content/early/2024/07/20/2024.07.17.603980.

Minhyeok Lee. Recent advances in deep learning for protein-protein interaction analysis: A comprehensive review. *Molecules*, 28(13), 2023. ISSN 1420-3049. doi: 10.3390/molecules28135169. URL https://www.mdpi.com/1420-3049/28/13/5169.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Hossam Nada, Yongseok Choi, Sungdo Kim, Kwon Su Jeong, Nicholas Meanwell, and Kyeong Lee. New insights into protein–protein interaction modulators in drug discovery and therapeutic advance. *Signal Transduction and Targeted Therapy*, 2024.

Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. Progen2: Exploring the boundaries of protein language models. *arXiv*, 2022.

NVIDIA. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models, 2025. URL https://arxiv.org/abs/2504.03624.

Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. Large-scale prediction of human protein–protein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, 9(10):4992–5001, 2010. doi: 10.1021/pr100618t.

Damiano Sgarbossa, Cyril Malbranke, and Anne Florence Bitbol. ProtMamba: a homology-aware but alignment-free protein state space model. *bioRxiv*, 2024. doi: 10.1101/2024.05.24.595730. URL https://www.biorxiv.org/content/early/2024/05/25/2024.05.24.595730.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L. Gable, Tao Fang, Nadezhda T. Doncheva, Sampo Pyysalo, Lars J. Jensen, and Christian von Mering. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 2023. doi: 10.1093/nar/gkac1000.

The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023. doi: 10.1093/nar/gkac1052.

Varun Ullanat, Bowen Jing, Samuel Sledzieski, and Bonnie Berger. Learning the language of protein-protein interactions. *Nature Communications*, 2026.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models, 2024. URL https://arxiv.org/abs/2406.07887.

Yingheng Wang, Zichen Wang, Gil Sadeh, Luca Zancato, Alessandro Achille, George Karypis, and Huzefa Rangwala. Long-context protein language model, 2024. URL https://arxiv.org/abs/2411.08909.

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A comprehensive and multi-task benchmark for protein sequence understanding. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

A PRIMA-H 650M

PRIMA-H 150M is the best model we tested, so we scale it and train PRIMA-H 650M. The training strategy is the same as the rest of the PRIMA family. PRIMA-H 650M keeps the family's design: $d_{\text{model}} = 1280$ over 33 blocks, 28 state-space and 5 attention at the same one-in-six spacing, a gated-MLP ratio of 2.0, and ∼643M parameters. The recipe is unchanged from Section 3.2, with a 25,000-token budget per device, GPUs × grad. accum. of 16, and a peak learning rate of 6×10^{-4} for pre-training and 2×10^{-4} for post-training. We evaluate it with the protocols of Sections 3.3 and 3.4. The results show that language modelling improves substantially, interaction modelling does not.

Held-out cross-entropy on the 181,883 single chains falls from 1.985 at 150M to 1.695, and perplexity improves in every length band (Table 6 of Appendix F). The comparison against ESM2 650M reproduces the patterns of Section 4.1: ESM2 is ahead below 1000 residues, the ordering reverses between 1000 and 1500, beyond 5000 residues PRIMA-H is 2.3× better.

On SKEMPI-R every metric moves the wrong way: Pearson goes from 0.451 to 0.417 (Figure 6 (a)), per-complex ρ from 0.152 to 0.121, margin over the null from 9.83% to 7.96%. On both pooled covered and uncovered rows, the absolute performance is worse and the gap is larger than at 150M. PRIMA-H 650M has similar performance to ESM2 650M, not better: it behaves like a good single-sequence protein language model. The ridge probe is inspected, and found healthy. However, on the classification benchmark Gold Standard PPI the model improves slightly, from 0.687 to 0.693 against per-model spreads of 0.003 (see Table 5 in Appendix E).

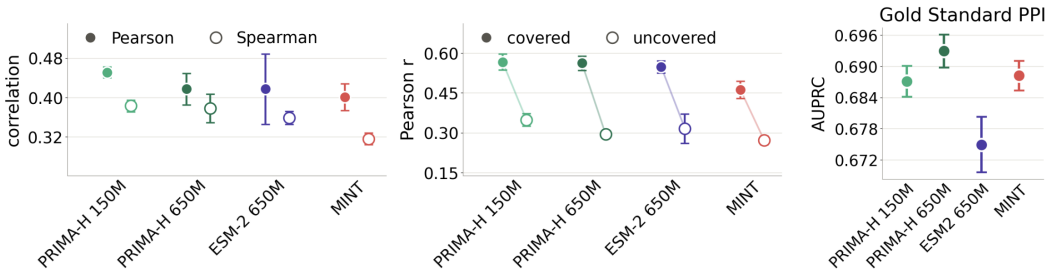


Figure 6: Results of PRIMA-H 650M compared to PRIMA-H 150M, ESM2 650M and MINT on (a) SKEMPI-R: Pearson and Spearman on out-of-fold pooled predictions; (b) SKEMPI-R Pearson on pooled covered and uncovered rows; (c) AUPRC on Gold Standard PPI

The explanation lies in how we test: in line with what MINT does, we always select the embeddings of the last layer of each model for all benchmarks. Instead, here we probe PRIMA-H 150M and 650M at different matching relative depths. This shows that both models peak at layer 9 (29% and 27% of their depth respectively) with pooled Pearson 0.488 and 0.420 on uncovered rows each: PRIMA-H is equally good at 150M as 650M, Figure 7. The best linearly decodable interaction information is not encoded in the top layers: this is expected, as the last layer is tuned for the MLM task. A disclaimer: layers are selected on the test folds here, so this describes the representations rather than a deployable protocol.

Scaling to 650M makes PRIMA-H a better language model that uses its partner less. In SKEMPI-R side A always has a mutated chain, and in 6,180 of the 6,706 rows only side A carries a mutation, so side B is the partner's reaction to a change it does not itself carry. Probing the two halves separately (Figure 7) shows what the extra capacity bought and what it cost: on side A the 650M is ahead of the 150M at almost every depth, on side B it is almost always behind at every depth. When using both side A and B together the two models' performance peak at the same layer and are indistinguishable: Pearson 0.488 on pooled rows, Pearson 0.420 on uncovered rows. This means that the extra capacity went into representing each chain rather than the relationship between them. By the last layer this stops being a wash and becomes a cost: adding the side-B half lowers the 650M's Pearson by 0.030, while leaving the 150M's essentially unchanged.

The training objective explains why the model does this. Masking residues on side A of the wild-type complexes, the 650M predicts them better than the 150M (1.62 against 1.79 nats), but it gains less from having side B present at all: removing the partner costs it 0.031 nats against the 150M's

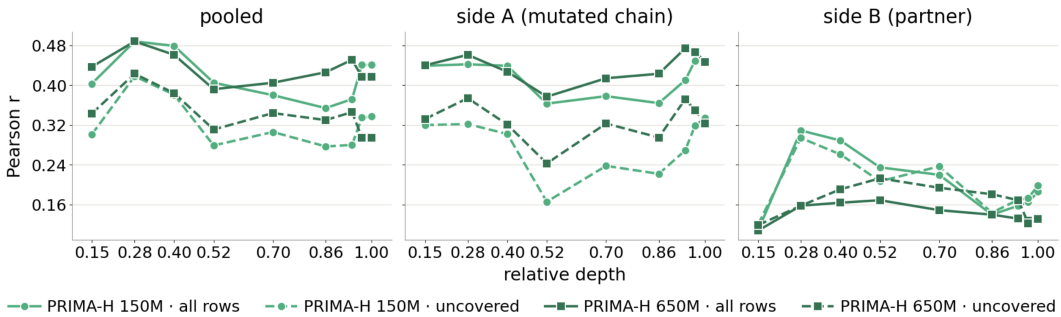


Figure 7: SKEMPI-R Pearson by depth for the two hybrids, probed with the protocol of Section 3.3 and sampled at matching relative depths. Side A is the always mutated chain and side B its partner, each probed alone. Values are means over the five partitions. Layers are chosen on the test folds, so this characterises the representations, not a deployable protocol.

0.044, and 0.015 against 0.031 when the masked residue is one of the positions SKEMPI mutates, which sit at the interface. A model can lower the masked-language loss either by learning each chain better or by learning how the two chains constrain each other, and the objective pays the same for both. Given more capacity, the larger model took the first route. However, we cannot rule out that longer training would change this, since PRIMA-H 650M sees about $4\times$ fewer tokens per parameter than PRIMA-H 150M.

B Implementation details

B.1 Architecture details

This section elaborates on Section 3.1. We build three PRIMA variants: PRIMA-M as a stack of state-space layers, PRIMA-A as a stack of self-attention layers, PRIMA-H as a hybrid stack of sequential state-space and self-attention layers. In order to have a parameter count for all variants as close as possible to 8M and 150M, and to have the least possible spread across the variants' size, we tune the number of layers, the embedding dimension and MLP ratio. This will result in the variants having different specifications. To test how further scale buys performance on PPI representation, we also train PRIMA-H 650M: see Appendix A.

- **PRIMA-M**: a stack of N bidirectional Mamba2 blocks, a token embedding layer, and an LM head tied to the embedding. Within each block the forward and reverse Mamba share their input and output projections, as in LC-PLM, and their outputs are summed; only the causal convolution and the SSM parameters are duplicated, so the reverse direction doubles the sequence-mixing compute while adding under 0.2% to the parameter count. We choose a bidirectional architecture because bidirectional encoders outperform causal models on representation tasks, and because a causal Mamba would give an asymmetric PPI representation: with the two chains concatenated, no position in chain A could ever see chain B, while every position in B sees all of A. Bidirectionality preserves $O(L)$ scaling.
- **PRIMA-A**: the same block wiring with every mixer replaced by full non-causal self-attention, computed with FlashAttention over packed variable-length sequences so padding is excluded. Contrary to state-space layers, attention has no notion of order and RoPE are mandatory. Because an attention mixer holds fewer parameters than a Mamba2 mixer at equal width, we widen the gated MLP to land on the same parameter count, rather than changing depth or model width.
- **PRIMA-H**: the PRIMA-M stack with one in every k interior blocks replaced by the PRIMA-A attention block; k is chosen to have around $15-20\%$ of attention layers, following the methods in NVIDIA (2025). The first and last blocks are always state-space, so the representation read by the LM head and by every downstream probe is produced by an SSM layer. Following hybrid language models that omit positional encodings on the grounds that their state-space layers supply position (Lieber et al., 2024), the attention layers in PRIMA-H use none. However, those models are causal and attention can recover order from the mask alone (Haviv et al., 2022); PRIMA-H is bidirectional, so attention without PE is permutation-equivariant and the state-space layers

Table 3: Training hyperparameters. Everything not listed is shared by every run: MLM with 15% masking, WSD schedule with 2% warmup and linear decay to 1% of the peak over the final 10% (pre-training) or 15% (post-training) of steps, AdamW with $\beta = (0.9, 0.95)$, weight decay 0.05 (pre-training) or 0.1 (post-training), gradient clipping 1.0, bf16, and token-budget batching capped at 256 sequences per device-batch. Per scale, the three variants differ only in the pre-training peak LR at 150M, tuned per variant.

scale	setting	pretrain (UniRef50, 10 ep)	post-train (STRING, 2 ep)
8M	token budget per device	450,000	450,000
	GPUs × grad. accum.	2	2
	peak LR	1×10^{-3}	3×10^{-4}
150M	token budget per device	62,000	62,000
	GPUs × grad. accum.	8	8
	peak LR	9×10^{-4} (M, H), 7×10^{-4} (A)	2×10^{-4}

are its only possible source of order. Cost is $O(L)$ in the state-space layers and $O(L^2)$ in the five attention layers, so PRIMA-H's cost curve lies between the two pure stacks and approaches PRIMA-A's only at lengths where the attention term dominates.

All three variants share the tokenizer, token embedding, tied LM head, pre-norm residual wiring with RMSNorm and fused add-norm, fp32 residual accumulation, and a gated MLP in every block; only the mixer and the width/depth adjustments above differ. Measured parameter counts are 153.2M (PRIMA-M), 153.2M (PRIMA-H) and 152.5M (PRIMA-A) at the 150M scale, and 7.8M, 8.2M and 7.7M at 8M. Further details in Table 1 of Section 3.1.

B.2 Training Strategy

- **Pre-training**: all PRIMA variants are pre-trained for 10 epochs, corresponding to ∼190B tokens. At the beginning of each epoch, a random sequence belonging to all the UniRef50 clusters is sampled from UniRef90. A random 0.5% subset of UniRef50 is held out as a validation set to monitor the training process. The learning rate (LR) schedule follows a warm-up - stable - decay (WSD) approach, with 2% warm-up and 10% linear decay to 1% of the peak LR value. The optimal peak LR is searched for every model, and chosen to be as close as possible at each scale to eliminate any possible confound. The peak LR for PRIMA-A 150M is 7×10^{-4}, smaller than the 9×10^{-4} used by PRIMA-M and PRIMA-H, as this resulted in a marginally lower loss. The weight decay is set to 0.05. Before the linear LR decay, PRIMA 8M variants showed clear signs of saturation, with both training and validation loss not having improved for the last 3 epochs; on the contrary, PRIMA 150M variants' loss was still decaying and they would benefit from longer training. No sign of overfitting is ever detected.
- **Post-training**: all PRIMA variants are post-trained for 2 epochs on 96M PPI pairs from STRING, corresponding to ∼165B tokens. The 96M PPI pairs are curated by the MINT team, and include exclusively physically interacting proteins. The training and validation split are also taken from MINT, with 250k PPI pairs held out as a validation set. All parameters of the models are unfrozen and updated. The LR schedule follows a WSD approach, with 2% warm-up and 15% linear decay to 1% of the peak LR value; the longer decay showed marginally better final loss. The weight decay is set to 0.1, twice as high as during pre-training to prevent memory loss. Before the linear LR decay, no PRIMA variants showed signs of saturation in either training or validation loss; the effect was more evident for the PRIMA 150M variants; whether they would benefit from longer training would require a dedicated analysis given the potential catastrophic forgetting. No sign of overfitting is ever detected. The sequences constituting a PPI pair are concatenated before being fed to the models, as described in Section 3.1.

Both pre-training and post-training implement a MLM loss task, following the BERT protocol and masking 15% of the input tokens. To ensure proper GPU utilization and minimize padding, batches are set by a fixed token budget, with maximum batch size of 256. The parameters are tuned to optimize the validation loss. The hyper-parameters per model are detailed in Table 3

C SKEMPI-R: DETAILS

This Appendix elaborates on Section 3.3. SKEMPI 2.0 (Jankauskaitė et al., 2019) is a standard benchmark for binding affinity: 6,706 point mutations across 345 complexes with $\Delta\Delta G$ acquired experimentally. A common protocol is to randomly assign complexes (complex-disjoint) to three folds following the approach in Jin et al. (2023), that the authors of MINT replicated. However, we find the protocol itself as well as the implementation from the MINT team have four defects.

- **MINT's split is an unseeded, unstratified shuffle.** MINT's pipeline randomly assigns complexes to three folds following the approach in Jin et al. (2023); however, the assignment is done with an unseeded shuffle, so the runs cannot be reproduced and results cannot be compared. Moreover, in the split we obtained, one fold's test $\Delta\Delta G$ spread is 1.40× narrower than its own training folds' and its mean is shifted by 0.404 kcal/mol. A correlation is bounded by the target's spread, so that fold depresses every model's score so much that it ends up determining the ranking of the models. Simulating the same procedure, a plain complex-disjoint shuffle produces a fold at least as damaged 45% of the time, meaning the protocol fails around half the time.
- **Complex-disjoint splitting leakage.** 72.9% of all MINT's test rows have a chain that is ≥90% identical to a chain seen when training the probe, and 41.4% have an exactly identical one. The clearest case: the four largest entries (1CHO, 3SGB, 1R0R, 1PPF; 1,133 rows, ∼17% of the benchmark) are four different serine proteases assayed against the same 51-residue inhibitor, so holding out one of those complexes results in hundreds of rows with a mutation at the same positions of an identical chain in training. The complexes are still different, but a simple lookup table that predicts a test row by the mean $\Delta\Delta G$ of the same mutation seen in a different complex in training (no model, no sequence) can answer from 17% to 46% of the benchmark depending on the fold. Scored on those rows exclusively, it reaches Pearson 0.479, inside the range that the six trained models achieve on the same rows. It is apparent that for a significant fraction of the benchmark the partner interaction does not matter, explaining how models trained on single sequences like ESM2 are still competitive. This is not accounted for in the MINT paper.
- **Per-fold correlations are averaged across incomparable scales.** MINT reports the mean correlation across the folds. However, the three folds have unhealthy $\Delta\Delta G$ spreads 45% of the time, so a fixed RMSE is worse than useless on one fold and respectable on another, and averaging correlations gives a narrow-spread fold a full one-third vote. Scoring within a fold also re-centres and re-scales inside it, which forgives a model whose predictions are on mutually inconsistent scales across folds. Also, a correlation is invariant to rescaling the prediction vector. Scoring each fold separately grants the model a separate intercept and slope in every fold, for free. Pooling grants it one, shared across all folds. A deployed model has one. Over the three folds we obtained with this approach, averaging per-fold Pearson rather than pooling the out-of-fold predictions inflates the performance of every model.
- **Pooled Pearson measures mostly the wrong variance.** 62% of $\Delta\Delta G$ variance in SKEMPI is within complex, so even a correctly pooled correlation is dominated by ranking complexes against one another rather than ranking mutations within a complex, which is one of the questions of interest.

For these reasons, we build a revised version of this benchmark: SKEMPI-R. We use the following approach:

- **Moment-matched folds.** The health of a fold is determined by three per-fold metrics: row count, $\sum(\Delta\Delta G)$ and $\sum(\Delta\Delta G)^2$. Matching those makes every fold have the same mean and spread as its own training set by construction. We start from a random complex-disjoint assignment and move one complex at a time, accepting moves that reduce the squared deviation of those three totals from their targets, with annealing. If successful, this should get a spread ratio close to 1 and the mean gap close to 0. Across five seeds our folds reach a spread ratio of 0.988–1.011, a mean gap of at most 0.003 kcal/mol, and row counts balanced to within 0.5%. The worst fold of the previous three-fold split sits at a spread ratio of 1.397 (its training data is 40% more spread out than the data it is tested on) and a mean gap of 0.404 kcal/mol. This strategy does not carry any information about which input maps to which label. All five partitions are seeded and recorded.
- **Five seeded partitions.** Every partition is saved with its seed. We check that different seeds give genuinely different splits rather than converging on one. The fraction of complexes sitting in the

same fold is 0.25-0.28 across the five seeds, against the 0.20 expected for unrelated splits. The five partitions are therefore close to independent draws. For each partition, all predictions are pooled into one vector to extract the correlation.

- **Four reported quantities.** We report both pooled Pearson and Spearman; the percentage improvement in RMSE over a do-nothing predictor that returns the training mean, reported in kcal/mol over the entire dataset; the mean per-complex Spearman, which measures within-complex ranking directly.
- **A memorisation split.** As mentioned, complex-disjoint splits still suffer from homologous chains leakage. To address this, we split the test rows by whether the same substitution, at the same position of a homologous chain, occurs anywhere in that fold's training rows (referred to as covered rows) and report every metric on both subsets, using the same folds and the same fitted probes. On the covered rows a model can score by recalling a value it has already seen; on the uncovered rows it cannot, so a margin that survives there is attributable to the representation rather than to recall. Every model loses ground between the two subsets, so this measures degree rather than drawing a clean line, and we report each model's drop alongside its scores. The uncovered rows are not free of homology in general: they are free only of this particular leak, since a mutated chain may still resemble a training chain while carrying different mutations.

To rank the models, we probe the last layer's frozen embeddings with ridge regression rather than an MLP. This is done for three main reasons. 1) Ridge measures the embeddings and what they encode linearly, not the probe; also, it has a closed form solution, so it brings no initialisation randomness. 2) SKEMPI has 6,706 data points, and given the embedding size of each model, the MLP probe used by MINT ends up having millions of parameters training on $\sim$4,500 rows: easy to overfit. 3) Ridge is fairer across models of different widths, has only one scalar penalty, and it does not add capacity to the probe. However, ridge is linear, and can measure only what can be linearly decoded from the embeddings; a better probe could extract more information if tuned properly. The spread we report comes from the different folds, not different initialisation seeds. Features are standardised with statistics from the training fold only; targets are left in kcal/mol, so predictions and RMSE need no back-transformation. For each model independently, the ridge penalty is chosen inside each training fold by grid search over 20 log-spaced values from 3.16 to 10^7, using an inner 5-fold split that is complex-disjoint like the outer folds. No selected penalty fell at either end of the grid. The results are showin in Table 4.

Table 4: SKEMPI-R, split by whether the same mutation appears on a homologous chain in that fold's training rows (1,763 covered and 4,943 uncovered of 6,706, averaged over partitions). Same folds and same fitted probes as Table 2; mean $\pm$ standard deviation over five seeded partitions. PRIMA-H 650M is reported as a scaling ablation and is excluded from the bolding, see Appendix A.

	Pearson		Spearman	
model	covered	uncovered	covered	uncovered
PRIMA-M 8M	0.409 ± 0.043	0.220 ± 0.059	0.394 ± 0.019	0.223 ± 0.029
PRIMA-H 8M	0.337 ± 0.041	0.208 ± 0.034	0.334 ± 0.018	0.215 ± 0.018
PRIMA-A 8M	0.332 ± 0.024	$\mathbf{0.245} \pm 0.066$	0.358 ± 0.014	$\mathbf{0.274} \pm 0.024$
ESM2 8M	$\mathbf{0.473} \pm 0.061$	0.228 ± 0.067	$\mathbf{0.432} \pm 0.027$	0.232 ± 0.029
PRIMA-M 150M	0.525 ± 0.029	0.255 ± 0.032	0.482 ± 0.025	0.267 ± 0.015
PRIMA-H 150M	$\mathbf{0.566} \pm 0.030$	$\mathbf{0.349} \pm 0.023$	0.515 ± 0.016	$\mathbf{0.318} \pm 0.024$
PRIMA-A 150M	0.485 ± 0.031	0.238 ± 0.041	0.461 ± 0.020	0.232 ± 0.037
ESM2 150M	0.562 ± 0.035	0.228 ± 0.038	$\mathbf{0.516} \pm 0.022$	0.291 ± 0.013
ESM2 650M	0.548 ± 0.022	0.316 ± 0.055	0.514 ± 0.021	0.281 ± 0.022
MINT 813M	0.533 ± 0.032	0.272 ± 0.017	0.462 ± 0.032	0.241 ± 0.012
PRIMA-H 650M	0.562 ± 0.027	0.294 ± 0.012	0.519 ± 0.038	0.304 ± 0.029

Around 40% of SKEMPI entries are not dimers: `ABC_DE` has five chains across two sides of one interface. In order to ensure a fair comparison across models, we follow MINT's encoding: glue each side into a single sequence, so every complex is presented as exactly two sides separated by one `<inter>` at the interface. This is the only input form all three model families can agree on (ESM2 has no separator token at all) and it is what we report. Concatenating each chain through `<inter>` buys marginal gains for PRIMA-H 150M over all pooled rows, and would make it harder

to compare with the other models tested. For every model, the extracted features consist in the difference between the wild-type and mutant embeddings of the same complex, computed with an identical readout on both. PRIMA encodes the two sides as one sequence in a single bidirectional forward pass (see Section 3.1), pools each side separately from that pass, and concatenates, giving $2 \times d_{model}$. MINT likewise sees both sides at once, through the cross-attention blocks of its ESM2 650M backbone, and pools each from that joint pass for 2×1280. ESM2 encodes each side in its own forward pass with the other absent, mean-pools, and concatenates. The last case has a consequence worth stating. In SKEMPI, a mutation usually lies on one side of the interface only. Because ESM2 encodes the two sides independently and has no way of propagating the interaction context, the unmutated side's embedding is bit-identical between the wild-type and the mutant, so half of its difference vector is exactly zero for those rows. PRIMA and MINT see the mutation on both sides, since each side's representation is computed in the presence of the other.

No sequence is cropped for any model: the longest SKEMPI complex is shorter than the per-chain limit MINT applies elsewhere. All models are read out at their final layer.

D SKEMPI: MINT'S PROTOCOL

This Appendix shows the performance of PRIMA over SKEMPI with under MINT's protocol. The probe follows their downstream setup. For each model, features are the difference between the wild-type and mutant embeddings taken from the final layer, with each side mean-pooled over residues excluding padding and the two sides concatenated ($2 \times d_{\text{model}}$). Features are standardised and the $\Delta\Delta G$ targets power-transformed, both fitted on the training fold alone. The probe is one hidden layer of width 640 with ReLU and dropout 0.2, trained with AdamW (weight decay 0.01) and batch size 16 on MINT's three complex-disjoint folds, selecting on Pearson. Each fold is fitted three times from different initialisations, and we report the mean and standard deviation across the three folds; this is the averaging that Section 3.3 identifies as a defect, reproduced here deliberately so the comparison is on MINT's terms.

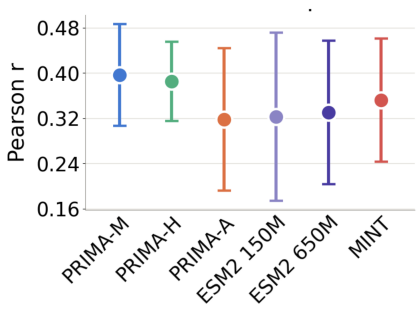


Figure 8: Results of PRIMA-M 150M, PRIMA-H 150M, PRIMA-A 150M, ESM2 150M, ESM2 650M and MINT on SKEMPI with MINT's protocol. An MLP of hidden size 640 is trained on top of each model's last layer embeddings. Results are Pearson correlation, mean and standard deviation over the folds.

E PPI CLASSIFICATION BENCHMARKS: DETAILS

This Appendix elaborates on the PPI classification benchmarks, Section 3.4. We use three binary classification benchmarks.

- **Gold Standard PPI** (Bernett et al., 2024): 163,019 train / 59,260 validation / 52,048 test pairs, class-balanced, reported as AUPRC. Built specifically to remove protein overlap, cross-split homology, and degree bias in negative sampling.
- **Human PPI** (Xu et al., 2022): 26,319 train / 234 validation / 180 test pairs, reported as accuracy.
- **Yeast PPI** (Xu et al., 2022): 4,945 train / 95 validation / 394 test pairs, reported as accuracy.

We evaluate binary PPI prediction on three datasets using the splits of MINT. Pair-level leakage is absent from all three. Comparing full-sequence hashes, no protein appears in more than one split of

any of the three datasets, no pair is duplicated across splits, and no evaluation row contains a protein seen in training. However, Human PPI and Yeast PPI are protein-disjoint re-splits of the PAN (Pan et al., 2010) and GUO (Guo et al., 2008) datasets, on which Bernett et al. (2024) demonstrated node-degree shortcuts. The re-split removes protein overlap but not the shortcut: with frozen ESM2 650M embeddings, a probe restricted to scoring each protein independently ($\text{logit} = g(a) + g(b)$, which cannot represent any interaction term) reaches 0.82 AUROC on Human PPI and outperforms a pairwise probe on Yeast PPI, 0.72 against 0.66. On the Gold Standard PPI the same probe is at chance, 0.52 against 0.69. We therefore treat Gold Standard PPI as the primary benchmark. We report the other two for comparability with prior work, but do not use them to rank models as they can be largely solved with no interaction knowledge.

We use MINT's probe to evaluate the models: an MLP with a single hidden layer of width 640 is trained on frozen embeddings from each model's last layer, with their hyperparameters, selecting on the validation split. Embeddings are mean-pooled over residues and special tokens are excluded. Because the hidden width is fixed while embedding dimension is not, the probe's first layer scales with the model's width: a model producing embeddings of dimension 1280 (e.g. ESM2 650M, MINT) will have a probe much larger than a model producing embeddings of dimension 640 (e.g. PRIMA 150M).

Table 5: PPI classification results for post-trained models. Gold Standard PPI is AUPRC; Human PPI and Yeast PPI are accuracy. Values are mean ± standard deviation over three probe-training repetitions; the embeddings, splits and test set are identical across repetitions, so the spread reflects probe initialisation and shuffling alone and not uncertainty from the test sample. Best per column within a scale in bold. PRIMA-H 650M is reported as a scaling ablation and is excluded from the bolding, see Appendix A.

model	Gold Standard PPI	Human PPI	Yeast PPI
PRIMA-M 8M	**0.671** ± 0.003	0.811 ± 0.018	0.591 ± 0.019
PRIMA-H 8M	0.670 ± 0.002	0.815 ± 0.007	0.592 ± 0.009
PRIMA-A 8M	0.659 ± 0.001	0.810 ± 0.014	**0.602** ± 0.005
ESM2 8M	0.663 ± 0.001	**0.821** ± 0.014	0.577 ± 0.010
PRIMA-M 150M	**0.689** ± 0.004	0.860 ± 0.007	0.619 ± 0.009
PRIMA-H 150M	0.687 ± 0.003	**0.887** ± 0.004	**0.645** ± 0.006
PRIMA-A 150M	0.675 ± 0.000	0.859 ± 0.011	0.620 ± 0.007
ESM2 150M	0.682 ± 0.004	0.859 ± 0.005	0.596 ± 0.010
ESM2 650M	0.675 ± 0.006	0.858 ± 0.011	0.620 ± 0.006
MINT 813M	0.688 ± 0.003	0.858 ± 0.011	0.630 ± 0.006
PRIMA-H 650M	0.693 ± 0.003	0.843 ± 0.033	0.602 ± 0.009

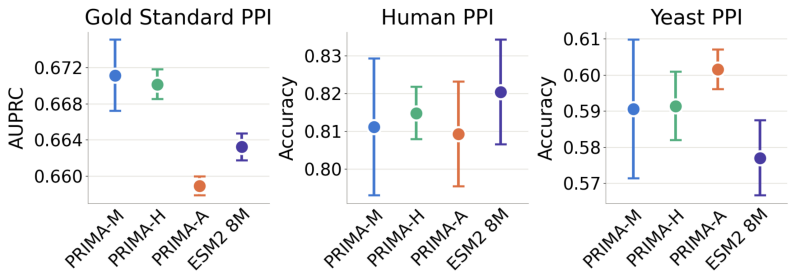


Figure 9: Results of PRIMA-M 8M, PRIMA-H 8M, PRIMA-A 8M and ESM2 8M on Gold Standard PPI, Human PPI and Yeast PPI

All three model families produce a $2 \times d_{model}$ readout, but they differ in whether the two chains are ever present in the same forward pass. PRIMA encodes the complex as one sequence in a single forward pass. Individual chains are mean-pooled separately excluding special tokens, and the two vectors concatenated, giving $2 \times d_{model}$ size. MINT also sees all chains at once: it encodes each with an ESM2 650M backbone and models inter-chain dependencies through cross-attention blocks, then pools each chain from that joint pass and concatenates, giving $2 \times 1280 = 2560$. We use its `sep_chains` readout, which its authors report as its best-performing configuration, and which matches PRIMA's readout construction exactly. ESM2 encodes each chain in its own forward pass

with the partner absent, mean-pooled, and the two vectors concatenated. Its per-chain representations are therefore not interaction-aware: any signal about the interaction has to be reconstructed by the probe from two independently computed vectors. This is the configuration MINT uses for its own ESM2 baselines. Sequences are not cropped for PRIMA or ESM2. MINT inputs are cropped at 4098 tokens per chain due to OOM constraints on a 80GB GPU, affecting 1.32% of Gold Standard PPI rows and no rows of Human PPI or Yeast PPI.

In line with MINT, we extract AUPRC for Gold Standard PPI and accuracy for Human and Yeast PPI. We report mean and standard deviation over 3 different initialisation seeds of the MLP probe.

F HELD-OUT SINGLE-SEQUENCE MODELLING: DETAILS

After pre-training, we evaluate every PRIMA variant on the single sequences from the STRING validation split, excluding all those that appear verbatim in the UniRef50 pre-training corpus. This leaves 181k sequences spanning 8-32,000 residues. The task mirrors the training objective: 15% of tokens are masked and perplexity is computed on the masked positions only, using the same masked positions for every model so that no part of the comparison is mask sampling noise. We report perplexity and stratify the results by input, since ESM2 was trained with a 1024-token cap. The goal of the test is to assess how well the models perform against single sequence PLMs like ESM2, and how well the attention layers can use the position information by state-space layers in the hybrid PRIMA-H, as no PE is directly fed at any point. Note that excluding verbatim matches does not exclude homologs: we sample UniRef90 members from UniRef50 clusters during pre-training, so a held-out chain may still be up to 50% identical to a training sequence.

Pooled over all 181,883 sequences, PRIMA 8M models (loss from 2.301 to 2.317) are ahead compared to ESM2 8M (loss of 2.346); conversely, PRIMA 150M models (loss from 1.978 to 1.992) are behind ESM2 150M (loss of 1.934). This is an indication of what already mentioned in Section 3.2. At 8M the PRIMA models (and likely ESM2 8M as well) are capacity-saturated, and the lower loss of the PRIMA family is an indication of a good combination of architecture and training recipe. The advantage of training ESM2 on $\sim$1T tokens against the $\sim$190B PRIMA used starts showing at 150M, where models are not capacity-saturated. However, the goal of pre-training PRIMA is not to build the best-in-class single sequence model, but rather to have it learn good representations that will transfer to the post-training phase. Moreover, 82.6% of the evaluation set is 1000 residues or shorter, inside the 1024-token window ESM2 was trained with, so the pooled figure largely reports who is better within that window. Splitting by length gives a different and more informative answer (Figure 1, Table 6).

Below 1000 residues ESM2 150M is ahead of every PRIMA variant, by 0.83 perplexity at $<$200 and 0.27 at 500-1000. Between 1000 and 1500 the ordering reverses, and from there the gap widens monotonically: 0.74 at 1500–2000, 1.43 at 2000–5000, and 5.62 above 5000 residues, where ESM2 150M reaches perplexity 10.28 against PRIMA-M's 4.66. The two families move in opposite directions. PRIMA's perplexity falls with length, as expected (longer chains give more context and more internal redundancy) while ESM2's reaches its minimum at 200–500 residues and rises thereafter. The crossover sits at ESM2's training cap. ESM2 degrades past its training length due to RoPE degradation outside the training length distribution, not the mixer, and PRIMA-A is the evidence. Although different from ESM2, PRIMA-A is a pure attention stack with RoPE trained without any length cap, and it tracks the two state-space variants across every bin. FlashAttention in PRIMA-A is what makes training without a cap affordable (Section 3.1).

The mixers differ slightly, and attention is last. At 8M the hybrid is best: -0.0089 ± 0.0004 nats against pure state-space and -0.0160 ± 0.0004 against pure attention. At 150M pure state-space is best, by 0.0067 ± 0.0007 over the hybrid, with pure attention last at 0.0137 ± 0.0008 behind. The ranking changes between scales, but pure attention is last at both. However, these gaps are small, a very small percentage of the overall loss.

As an ablation on scaling, the results for PRIMA-H 650M are also reported in Table 6. The observed pattern is the same, and it strengthens the hypothesis that the PRIMA-H 650M is not capacity saturated, and that the advantage of ESM2 650M lies in 5$\times$ pre-training budget.

Table 6: Mean masked-token perplexity by sequence length, on 181,883 held-out single chains with no UniRef50 overlap and no length cap. Lower is better; the best in each column within a scale is bold. The 650M block holds only the hybrid and ESM2, since PRIMA-M and PRIMA-A were not trained at that scale. The rightmost column is the pooled held-out cross-entropy in nats.

	<200	200–500	500–1k	1k–1.5k	1.5k–2k	2k–5k	>5k	pooled CE
sequences	77,783	47,412	25,094	19,939	6,086	5,201	368	181,883
PRIMA-M 8M	11.55	10.36	10.24	10.10	9.90	9.19	6.87	2.310
PRIMA-H 8M	**11.51**	**10.29**	**10.16**	**9.98**	**9.73**	**8.96**	**6.31**	**2.301**
PRIMA-A 8M	11.63	10.43	10.33	10.14	9.93	9.22	6.84	2.317
ESM2 8M	11.94	10.45	10.40	10.28	10.36	11.26	13.54	2.346
PRIMA-M 150M	9.20	7.54	7.31	**7.33**	**7.25**	**6.72**	4.66	1.978
PRIMA-H 150M	9.26	7.58	7.34	7.38	7.30	6.74	**4.63**	1.985
PRIMA-A 150M	9.21	7.65	7.43	7.48	7.39	6.85	4.77	1.992
ESM2 150M	**8.37**	**6.91**	**7.04**	7.63	7.99	8.15	10.28	**1.934**
PRIMA-H 650M	7.00	5.94	5.83	5.84	**5.79**	**5.35**	**3.75**	1.695
ESM2 650M	**6.16**	**5.18**	**5.26**	**5.61**	6.08	6.41	8.67	**1.613**

G EFFICIENCY AND SCALING BENCHMARK: DETAILS

We measure the cost of a single forward pass, with latency and peak GPU memory, over input lengths from 128 to 65,536 tokens on one NVIDIA H100 80GB. We report batch size 1 and 8, the largest batch for which every cell completes for every model and at which all models are within 1% of their peak throughput. MINT consumes a chain pair, so an input of length L translates as two chains of $L/2$. Every model runs in bf16, the dtype PRIMA is trained in, with the fastest attention kernel available to it (FlashAttention for PRIMA-A and PRIMA-H, PyTorch's block-wise `scaled_dot_product_attention` for ESM2, and its own implementation for MINT). A block-wise kernel never materialises the $L \times L$ score matrix, so ESM2's memory grows linearly in length and only its computational cost stays quadratic; measuring attention against a naïve implementation would overstate the comparison. Passes are bracketed by `torch.cuda.synchronize()`, we discard five warm-ups, and we report the median of between 5 and 50 timed passes. One model is tested on the GPU at a time. Passes that exhaust memory are recorded as such.

We observed one issue with the mamba-ssd implementation when processing large inputs. Mamba-ssm's SSD kernel offsets its `dt` pointer by `pid_b` $\times$ `stride_dt_batch` in 32-bit arithmetic, and `dt` views the block's input projection, so a forward pass is safe only while $d_{\text{in_proj}} \cdot B \cdot L < 2^{31}$. This is roughly $759{,}902$ tokens at 150M ($d_{\text{in_proj}} = 2826$) and $1{,}667{,}300$ at 8M (1288), tightening as the model widens. At $L = 32{,}768$ a batch of 23 reproduces a chunked reference exactly and a batch of 24 does not; in bf16 the pass completes, raises nothing, and returns one corrupted row out of 24. The limit is in the implementation, not the architecture: a bare linear layer of the same shape is bit-exact, PRIMA-A has no state-space layers and a wider projection yet computes the same cell correctly. The issue is reported at `https://github.com/state-spaces/mamba/issues/686`; the analogous overflow in the Mamba-1 CUDA kernels, `https://github.com/state-spaces/mamba/issues/880`, is fixed. A single 64-bit cast in the Triton kernel would lift it. The failure can be silent; no reported number is affected.

Table 7: Inference cost on one H100 80GB, bf16, with each model's fastest available attention kernel. Latency is per sequence; memory is the peak allocated for the whole batch. OOM marks a cell that exhausted the 80 GB device; † marks a cell refused in advance by the memory guard rather than run to failure. Token counts include each family's special tokens and are the quantity the fits in Table 8 use.

	8M scale				150M scale				larger baselines	
L	M	H	A	ESM2	M	H	A	ESM2	ESM2-650M	MINT
latency, ms per sequence — batch size 1										
512	22.0	21.5	9.1	3.5	77.0	72.8	31.4	14.8	16.4	24.1
2,048	21.8	21.3	9.1	3.8	78.3	73.6	32.3	18.8	27.8	99.7
8,192	23.4	22.9	10.0	24.2	79.7	74.0	37.0	127.6	174.8	1265.6
16,384	23.4	22.9	12.8	87.0	78.9	73.8	102.1	452.0	571.1	10546.2
32,768	23.5	23.2	39.7	334.6	114.2	144.4	329.5	1708.6	2068.7	OOM
65,536	23.6	50.6	141.0	1316.0	221.0	366.8	1147.9	6655.8	7839.4	OOM
latency, ms per sequence — batch size 8										
512	2.9	2.8	1.2	0.5	9.8	9.2	4.0	2.1	3.9	9.2
2,048	2.9	2.9	1.2	1.9	9.8	9.2	5.7	11.9	19.3	93.3
8,192	3.0	3.1	4.0	22.4	27.4	26.9	32.9	120.3	164.7	OOM
16,384	4.9	6.6	11.4	85.0	53.9	58.4	96.8	441.6	562.1	OOM†
32,768	9.5	16.3	38.0	330.9	107.1	137.4	320.1	1690.9	2062.5	OOM†
65,536	18.9	46.0	136.4	1313.8	214.5	362.0	1148.7	6659.5	7913.2	OOM†
peak memory, GB — batch size 1										
512	0.06	0.06	0.05	0.06	0.36	0.36	0.35	0.35	1.37	1.76
2,048	0.07	0.07	0.07	0.08	0.40	0.40	0.39	0.40	1.48	2.89
8,192	0.15	0.15	0.13	0.16	0.57	0.57	0.53	0.60	1.89	20.82
16,384	0.24	0.25	0.21	0.27	0.80	0.80	0.72	0.88	2.45	77.97
32,768	0.44	0.44	0.37	0.49	1.26	1.26	1.10	1.42	3.57	OOM
65,536	0.83	0.83	0.69	0.94	2.18	2.18	1.87	2.51	5.80	OOM
peak memory, GB — batch size 8										
512	0.10	0.10	0.09	0.10	0.46	0.46	0.43	0.44	1.55	2.42
2,048	0.25	0.25	0.21	0.26	0.81	0.81	0.72	0.77	2.21	11.38
8,192	0.83	0.83	0.69	0.90	2.19	2.19	1.87	2.07	4.83	OOM
16,384	1.61	1.61	1.33	1.74	4.02	4.02	3.40	3.81	8.33	OOM†
32,768	3.17	3.17	2.61	3.43	7.70	7.70	6.45	7.30	15.32	OOM†
65,536	6.28	6.29	5.17	6.82	15.05	15.05	12.57	14.26	29.30	OOM†

Table 8: Fitted cost scaling, cost $\propto$ (tokens)^k, at batch size 8 over $L \geq 16{,}384$ — the region where the GPU rather than host-side dispatch is the bottleneck. Time recovers the architectures' predicted behaviour; memory is linear for every model, because ESM2 is measured with a block-wise attention kernel that never materialises the score matrix.

scale	model	time exponent	memory exponent
8M	PRIMA-M	0.98	0.98
	PRIMA-H	1.40	0.98
	PRIMA-A	1.79	0.98
	ESM2 8M	1.98	0.98
150M	PRIMA-M	1.00	0.95
	PRIMA-H	1.32	0.95
	PRIMA-A	1.78	0.94
	ESM2 150M	1.96	0.95
	ESM2 650M	1.91	0.91