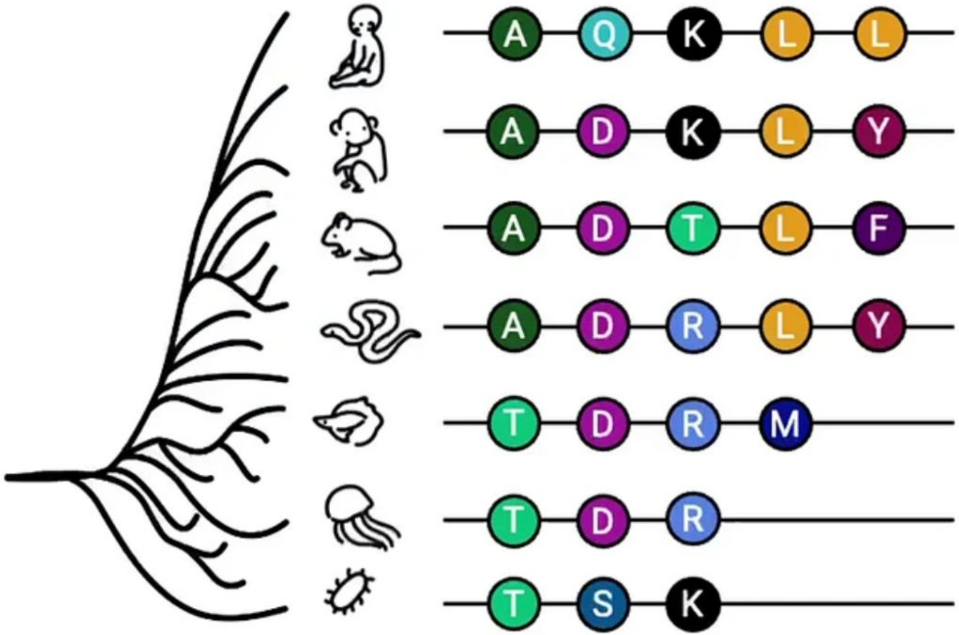
Protriever: End-to-End Differentiable Protein Homology Search for Fitness Prediction

Ruben Weitzman, Peter Mørch Groth, Lood Van Niekerk, Aoi Otani, Yarin Gal, Debora S. Marks, Pascal Notin

Gökçe Uludoğan
PhD Candidate

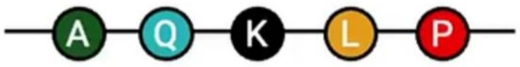
LifeLU Reading Group
4 September 2025

Protein function emerges from **evolutionary constraints** shaped over billions of years



- Homologous sequences reveal which amino acid positions are tolerant or not to mutation
- Model learns from evolution to score sequence with respect to it
- Allows variant effect prediction by comparing likelihood to wild type

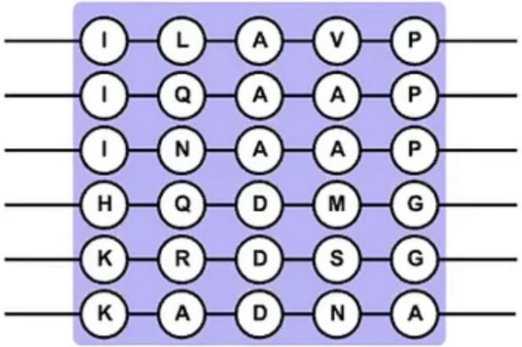
$p(\mathbf{x})$



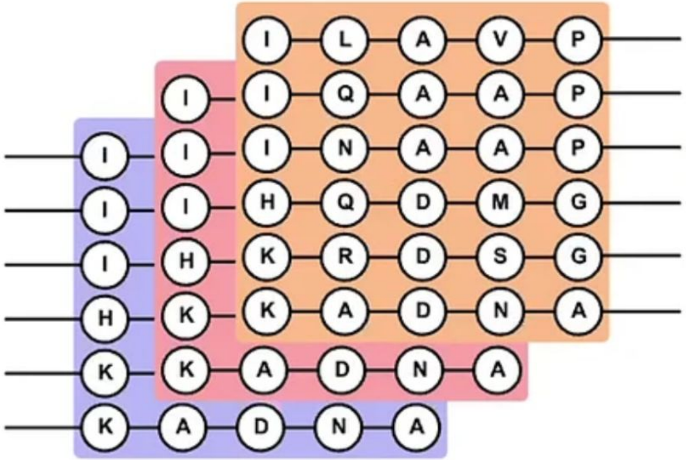
$$\rightarrow \log\left(\frac{p(\mathbf{x}_{\text{var}})}{p(\mathbf{x}_{\text{ref}})}\right)$$

https://icml.cc/virtual/2025/poster/45825

Sequence homology is crucial for proper representation learning of proteins



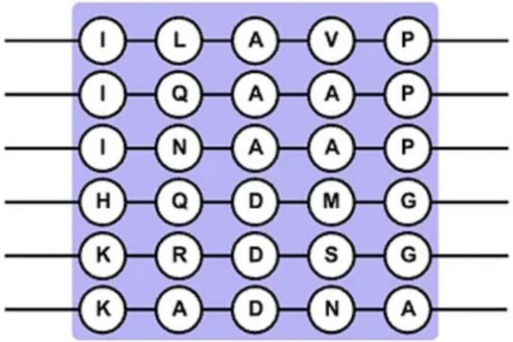
Multiple Sequence Alignment (MSA) routine returns sequences based on sequence similarity, by aligning to a query



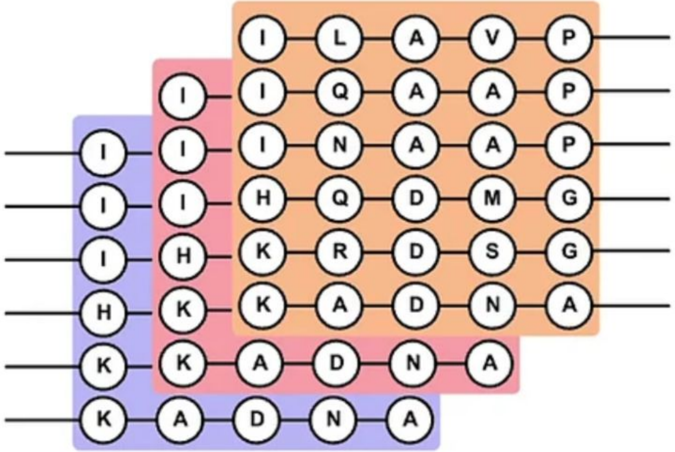
Learning across protein families, whether aligned (MSA Transformer) or not (PoET) yields best fitness and structure prediction results

https://icml.cc/virtual/2025/poster/45825

Sequence homology is crucial for proper representation learning of proteins



Multiple Sequence Alignment (MSA) routine returns sequences based on sequence similarity, by aligning to a query



Learning across protein families, whether aligned (MSA Transformer) or not (PoET) yields best fitness and structure prediction results

Downsides:

- Slow as requires dynamical programing alignments of large number of sequences
- Not differentiable and adapted to PLM paradigm
- 2 step process: retrieval done independently of the downstream task

https://icml.cc/virtual/2025/poster/45825

Traditional Protein Modeling

- Two stage Pipeline
  - **MSA retrieval:** retrieving homologs via Multiple Sequence Alignments (MSAs)
  - **Sequence Modeling:** training models on one or more of these alignments.
- Limitations
  - **Misses distant homologs** that fall below alignment thresholds.
  - Struggles with sequences containing **large insertions, deletions, or rearrangements.**
  - Operates **independently of modeling goals**, relying on heuristic alignments.
  - Requires **separate MSAs and models** for new families -> computationally costly, not scalable.

Protein Language Models as Alignment-free Alternatives

- Large-scale pLMs enable **flexible, alignment-free** use of diverse protein sequences
- Single-sequence models **underperform family-specific methods** for variant effect prediction, especially on **rare/specialized proteins**
- Hybrid approaches combine **pLMs with family-specific context** for improved performance.

Hybrid Strategies for pLMs + Evolutionary Context

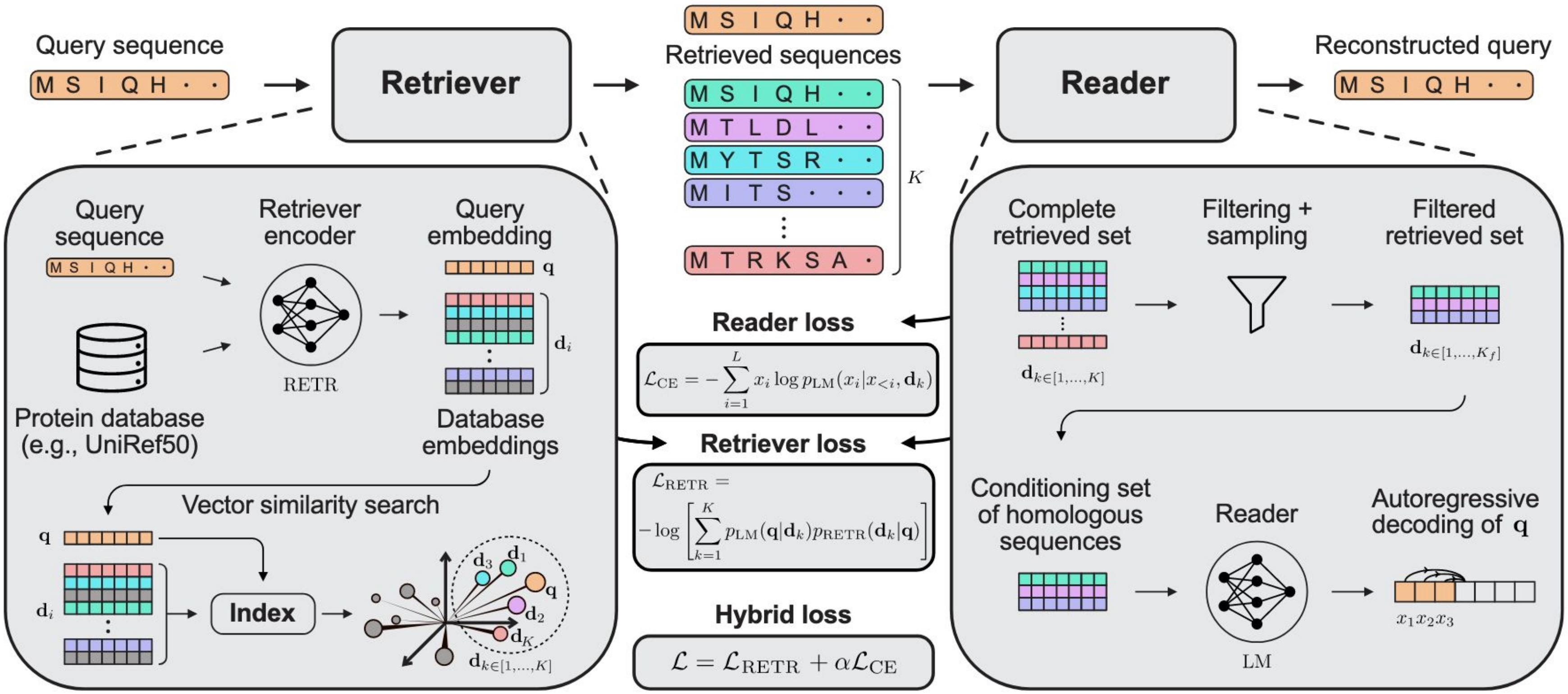
- **Fine-tuning on family sequences:** Evotuning (UniRep), spiked fine-tuning (ESM-1v)
- **Training across homolog sets:**
  - Entire MSAs (MSA Transformer, MSA Pairformer)
  - Concatenated homologous sequences (PoET, ProtMamba)
- **Retrieval at inference:** T
  - Combining unconditional LM with MSA position-specific frequencies (Tranception)
  - Integrating dependencies across MSA positions (TranceptEVE)
- **Limitation:** All rely on **static, similarity-based homology sets** -> *models cannot refine or backprop through retrieval.*

What if finding the right protein homologs wasn't a slow search, but a learned part of the model itself?

What if finding the right protein homologs wasn't a slow search, but a learned part of the model itself?

> **Protriever:** End-to-End Differentiable Protein Homology Search

Protriever


Query sequence
M S I Q H · ·
Retriever
Retrieved sequences
M S I Q H · ·
M S I Q H · ·
M T L D L · ·
M Y T S R · ·
M I T S · · ·
⋮
M T R K S A ·
K
Reader
Reconstructed query
M S I Q H · ·
Query sequence
M S I Q H · ·
Protein database (e.g., UniRef50)
Retriever encoder
RETR
Query embedding
$\mathbf{q}$
$\mathbf{d}_i$
Database embeddings
Vector similarity search
$\mathbf{q}$
$\mathbf{d}_i$
Index
$\mathbf{d}_1$
$\mathbf{d}_2$
$\mathbf{d}_3$
$\mathbf{d}_K$
$\mathbf{q}$
$\mathbf{d}_{k\in[1,\dots,K]}$
Reader loss
$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{L} x_i \log p_{\text{LM}}(x_i|x_{<i}, \mathbf{d}_k)$
Retriever loss
$\mathcal{L}_{\text{RETR}} = -\log\left[\sum_{k=1}^{K} p_{\text{LM}}(\mathbf{q}|\mathbf{d}_k)p_{\text{RETR}}(\mathbf{d}_k|\mathbf{q})\right]$
Hybrid loss
$\mathcal{L} = \mathcal{L}_{\text{RETR}} + \alpha\mathcal{L}_{\text{CE}}$
Complete retrieved set
$\mathbf{d}_{k\in[1,\dots,K]}$
Filtering + sampling
Filtered retrieved set
$\mathbf{d}_{k\in[1,\dots,K_f]}$
Conditioning set of homologous sequences
Reader
LM
Autoregressive decoding of $\mathbf{q}$
$x_1 x_2 x_3$

Protriever

- **Components:** Retriever, Index, Reader

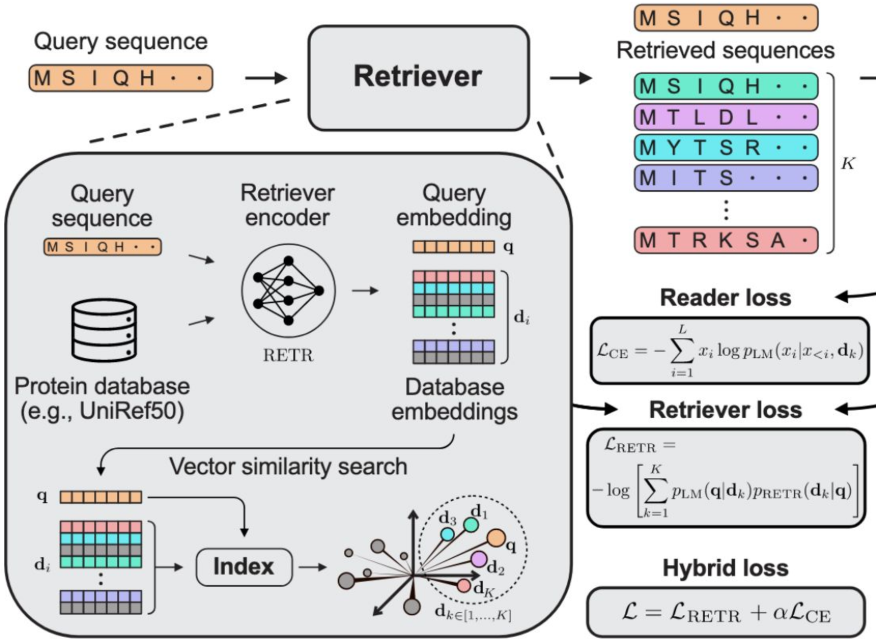
- **Process:**
    - Retriever searches fixed index of sequence embeddings for homologs
    - Reader **conditions on retrieved sequences** to perform target task

- **Training:**
    - **Self-supervised** (e.g., autoregressive decoding)
    - Reader provides **gradient feedback ➔** retriever refines embedding space

Retriever

- **Architecture**:
  - Transformer encoder initialized with ESM-2 (35M)
  - Last-layer average pooling ➔ 480-di embeddings
  - Similarity via cosine score

- **Pretraining**
  - Dense Passage Retrieval (DPR): homologs closer than non-homologs in embedding space



Query sequence
M S I Q H · ·
Retriever
M S I Q H · ·
Retrieved sequences
M S I Q H · ·
M T L D L · ·
M Y T S R · ·
M I T S · · ·
⋮
M T R K S A ·
K
Query sequence
M S I Q H · ·
Retriever encoder
Query embedding
$\mathbf{q}$
$\mathbf{d}_i$
RETR
Protein database (e.g., UniRef50)
Database embeddings
Vector similarity search
$\mathbf{q}$
$\mathbf{d}_i$
Index
$\mathbf{d}_3$ $\mathbf{d}_1$
$\mathbf{q}$
$\mathbf{d}_2$
$\mathbf{d}_K$
$\mathbf{d}_{k\in[1,\ldots,K]}$
Reader loss
$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{L} x_i \log p_{\mathrm{LM}}(x_i|x_{<i}, \mathbf{d}_k)$
Retriever loss
$\mathcal{L}_{\mathrm{RETR}} = -\log \left[\sum_{k=1}^{K} p_{\mathrm{LM}}(\mathbf{q}|\mathbf{d}_k) p_{\mathrm{RETR}}(\mathbf{d}_k|\mathbf{q})\right]$
Hybrid loss
$\mathcal{L} = \mathcal{L}_{\mathrm{RETR}} + \alpha\mathcal{L}_{\mathrm{CE}}$

Retriever Pretraining

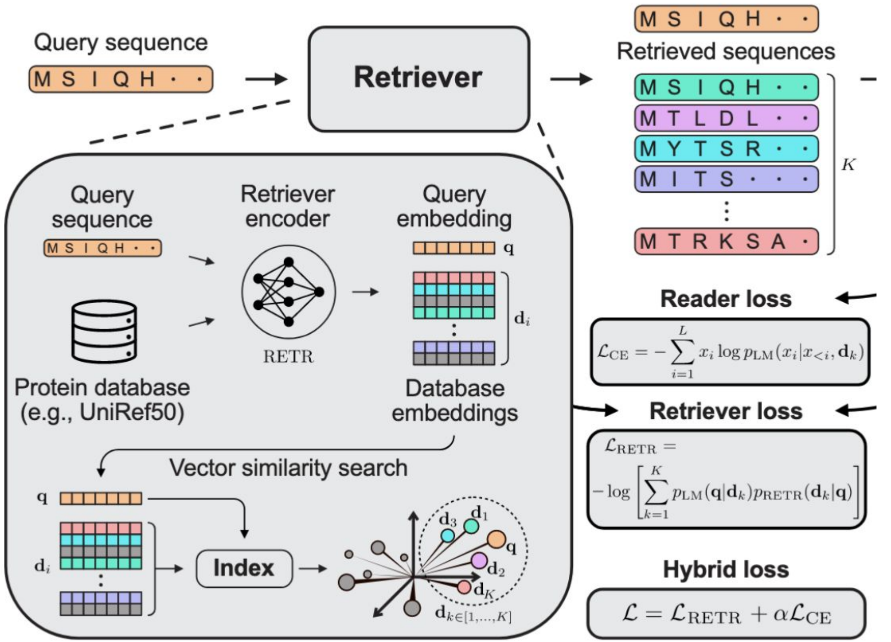
Training data: UniRef50 homologs via BLAST all-vs-all

Loss: negative log-likelihood over positives vs. negatives

Negatives: random + in-batch positives

Sampling: inverse-weighting by cluster size (UniRef50 → UniRef100)

Augmentation: random sequence reversal (N→C / C→N)


Query sequence
M S I Q H · ·
Retriever
M S I Q H · ·
Retrieved sequences
M S I Q H · ·
M T L D L · ·
M Y T S R · ·
M I T S · · ·
M T R K S A ·
K
Query sequence
M S I Q H · ·
Retriever encoder
RETR
Query embedding
q
d_i
Protein database (e.g., UniRef50)
Database embeddings
Vector similarity search
q
d_i
Index
d_3
d_1
q
d_2
d_K
d_{k∈[1,...,K]}
Reader loss
$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{L} x_i \log p_{\mathrm{LM}}(x_i|x_{<i}, \mathbf{d}_k)$
Retriever loss
$\mathcal{L}_{\mathrm{RETR}} = -\log\left[\sum_{k=1}^{K} p_{\mathrm{LM}}(\mathbf{q}|\mathbf{d}_k)p_{\mathrm{RETR}}(\mathbf{d}_k|\mathbf{q})\right]$
Hybrid loss
$\mathcal{L} = \mathcal{L}_{\mathrm{RETR}} + \alpha\mathcal{L}_{\mathrm{CE}}$

$$\mathcal{L}_{\text{pretrain}} = -\sum_{i=1}^{M}\sum_{j=1}^{m_i}\log\frac{e^{s\left(\mathbf{q}_i,\mathbf{d}^{+}_{i,j}\right)}}{e^{s\left(\mathbf{q}_i,\mathbf{d}^{+}_{i,j}\right)}+\sum_{k=1}^{n}e^{s\left(\mathbf{q}_i,\mathbf{d}^{-}_{i,k}\right)}}$$

Index Construction & Search

- **Index**: ≈62M UniRef50 embeddings (retriever-encoded), stored in **Faiss**

- **Staleness issue**: retriever updates → embeddings outdated
  Re-encode full index 10× during training for balance

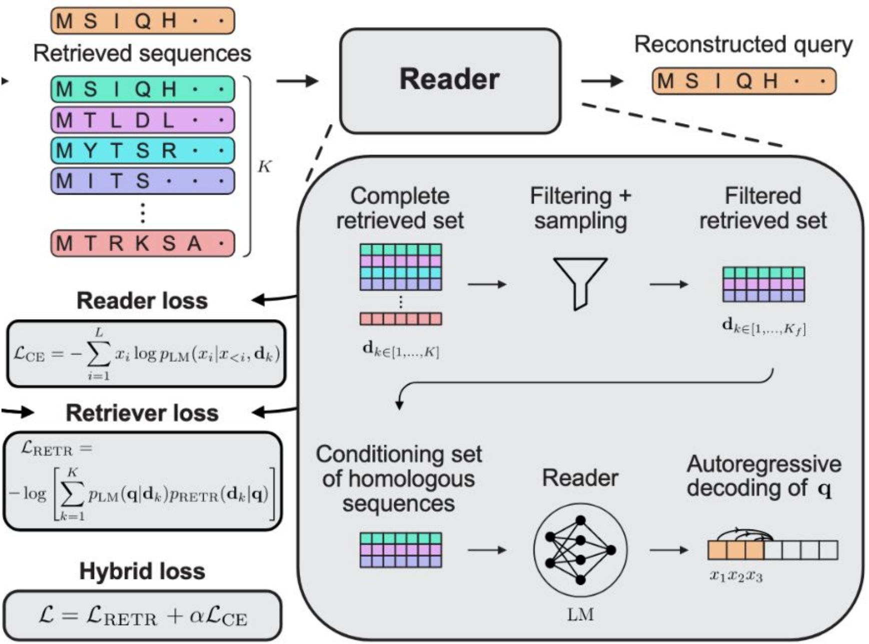
- **Efficiency optimizations**:
  IVF: k-means partitions; search limited to nearest clusters
  PQ: compress vectors → lower memory, retain accuracy

- **Scalability**: distributed across multiple GPUs; queries handled in parallel and aggregated

Reader

- **Backbone**: PoET (Protein Evolutionary Transformer)
  - Operates on concatenated sets of homologs (no alignment required)
  - Captures interactions across query + retrieved sequences
- **Initialization**: pretrained on UniRef50 ➔ strong foundation for end-to-end training
- **Flexibility**: adaptable to other architecture (e.g., Fusion-in-Decoder)

  Advantage: learns evolutionary context directly from retrieved sets



M S I Q H · ·
Retrieved sequences
M S I Q H · ·
M T L D L · ·
M Y T S R · ·
M I T S · · ·
⋮
M T R K S A ·
K
Reader
Reconstructed query
M S I Q H · ·
Reader loss
$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{L} x_i \log p_{\mathrm{LM}}(x_i | x_{<i}, \mathbf{d}_k)$
Retriever loss
$\mathcal{L}_{\mathrm{RETR}} = -\log \left[\sum_{k=1}^{K} p_{\mathrm{LM}}(\mathbf{q}|\mathbf{d}_k) p_{\mathrm{RETR}}(\mathbf{d}_k|\mathbf{q}) \right]$
Hybrid loss
$\mathcal{L} = \mathcal{L}_{\mathrm{RETR}} + \alpha \mathcal{L}_{\mathrm{CE}}$
Complete retrieved set
$\mathbf{d}_{k \in [1,\ldots,K]}$
Filtering + sampling
Filtered retrieved set
$\mathbf{d}_{k \in [1,\ldots,K_f]}$
Conditioning set of homologous sequences
Reader
LM
Autoregressive decoding of $\mathbf{q}$
$x_1 x_2 x_3$

Protriever Training

- **Goal**: retriever learns from reader feedback ➜ rank useful homologs closer to query
- **Relevance score**: softmax over top-K retrieved sequences

$$p_{\mathrm{RETR}}(\mathbf{d} \mid \mathbf{q}) = \frac{\exp(s(\mathbf{d}, \mathbf{q})/\theta)}{\sum_{k=1}^{K} \exp\left(s\left(\mathbf{d}_k, \mathbf{q}\right)/\theta\right)}$$

- **End-to-end training of Multi-Document Reader and Retriever (EMDR)**
  - Treats retrieved sequences as latent variables
  - Combines reader likelihood * pRETR
  - Stop-gradient on reader ➜ only retriever updated

$$\mathcal{L}_{\mathrm{EMDR}} = -\log\left[\sum_{k=1}^{K} p_{\mathrm{LM}}\left(\mathbf{q} \mid \mathbf{d}_k\right) p_{\mathrm{RETR}}\left(\mathbf{d}_k \mid \mathbf{q}\right)\right]$$

- **Alternatives tested**: Perplexity Distillation (PDist), LOOP

Fitness prediction with Protriever

Reader training: conditional autoregressive LM, conditioned on K retrieved sequences

$$P(x) = P_{\mathrm{RETR}}(\mathcal{D}_K|x) \prod_{i=1}^{l} P_{\mathrm{LM}}(x_i|x_{<i}, \mathcal{D}_K).$$

Fitness score: log-likelihood ratio of mutant vs. wild-type

$$F_x = \log \frac{P\left(x^{\mathrm{mut}}\right)}{P\left(x^{\mathrm{wt}}\right)}.$$

With shared retrieval set:

$$F_x = \log \frac{P_{\mathrm{LM}}\left(x^{\mathrm{mut}}|\mathcal{D}_K\right)}{P_{\mathrm{LM}}\left(x^{\mathrm{wt}}|\mathcal{D}_K\right)}.$$

Indexing:

- Encode all ~62M UniRef50 seqs (4×A100 GPUs, ~2h with FlashAttention)
- Build Faiss index in 3–4 min

Fitness prediction with Protriever

Retrieval & Re-ranking: diversity maximization ➜ broaden evolutionary coverage

Inference-time sampling:

- Filter homologs <15% similarity
- Sample 2,560 UniRef100 seqs (weighted by inverse UniRef90 cluster size)
- Encode + cluster (k=50) ➜ sample clusters with weight
- Vary parameters (a,T) ➜ 5 diversity/relevance trade-offs

$$\sqrt{s} \cdot \left(1 + \mathrm{e}^{-ad/T}\right)^{-1},$$

Ensembling:

- Conditioning set sizes: 6k, 12k, 24k tokens
- 5 diversity strategies × 3 set sizes = 15 forward passes
- Robust estimates via diversity + length ensembling

Evaluation: ProteinGym Substitution Benchmark

Dataset: 217 deep mutational scanning (DMS) assays

- Measure functional effects of single amino acid substitutions
- Provide comprehensive fitness landscapes

Challenge: detect subtle biochemical effects from minor sequence changes

Metrics:

- **Global**: Spearman, AUC, MCC → overall mutation effect prediction
- **Top-end**: NDCG, top-K recall → most relevant for protein design

Setup:

- **Zero-shot:** score all DMS sequences using retrieved homologs
- Bidirectional scoring (N→C, C→N) improves predictions

Protriever achieves the best performance across all metrics

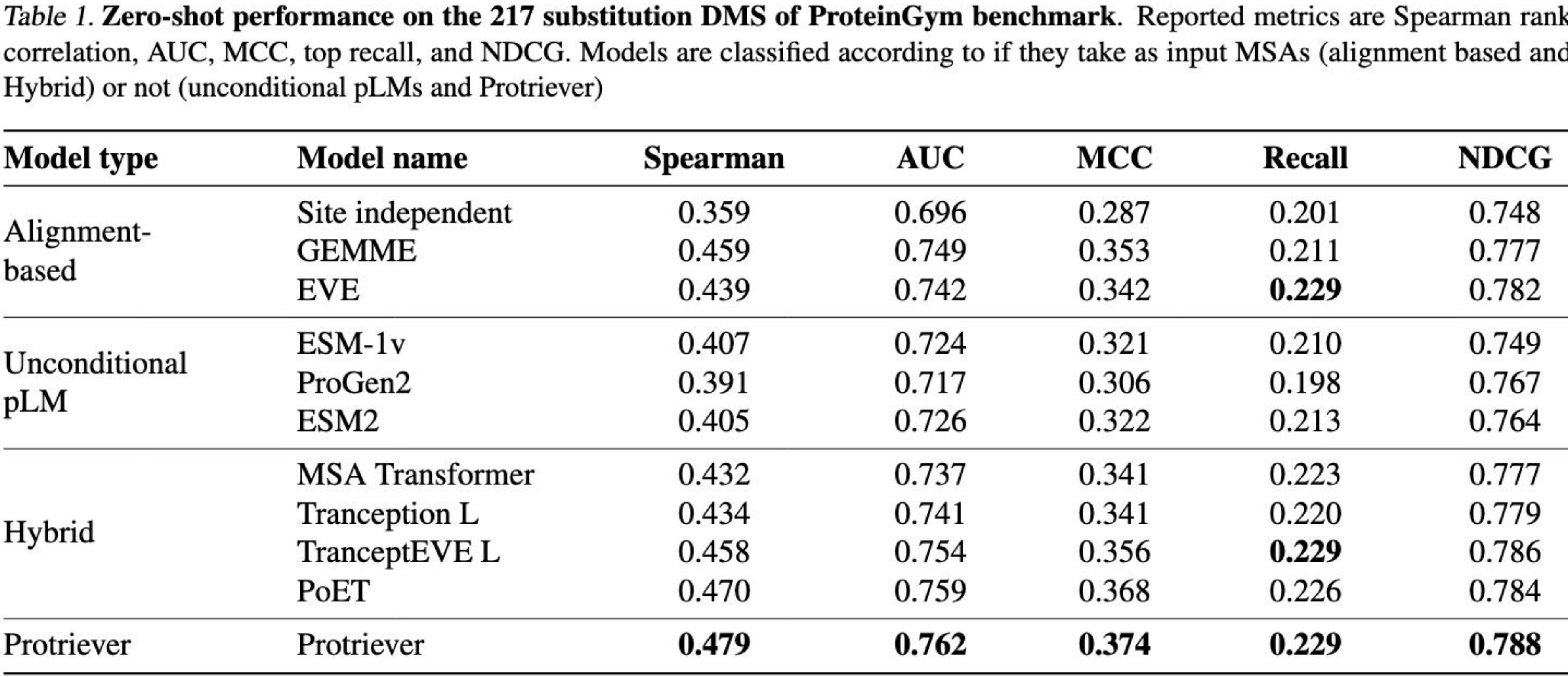


Table 1. **Zero-shot performance on the 217 substitution DMS of ProteinGym benchmark**. Reported metrics are Spearman rank correlation, AUC, MCC, top recall, and NDCG. Models are classified according to if they take as input MSAs (alignment based and Hybrid) or not (unconditional pLMs and Protriever)

Model type	Model name	Spearman	AUC	MCC	Recall	NDCG
Alignment-based	Site independent	0.359	0.696	0.287	0.201	0.748
	GEMME	0.459	0.749	0.353	0.211	0.777
	EVE	0.439	0.742	0.342	**0.229**	0.782
Unconditional pLM	ESM-1v	0.407	0.724	0.321	0.210	0.749
	ProGen2	0.391	0.717	0.306	0.198	0.767
	ESM2	0.405	0.726	0.322	0.213	0.764
Hybrid	MSA Transformer	0.432	0.737	0.341	0.223	0.777
	Tranception L	0.434	0.741	0.341	0.220	0.779
	TranceptEVE L	0.458	0.754	0.356	**0.229**	0.786
	PoET	0.470	0.759	0.368	0.226	0.784
Protriever	Protriever	**0.479**	**0.762**	**0.374**	**0.229**	**0.788**

Protriever performance is consistent across different protein families

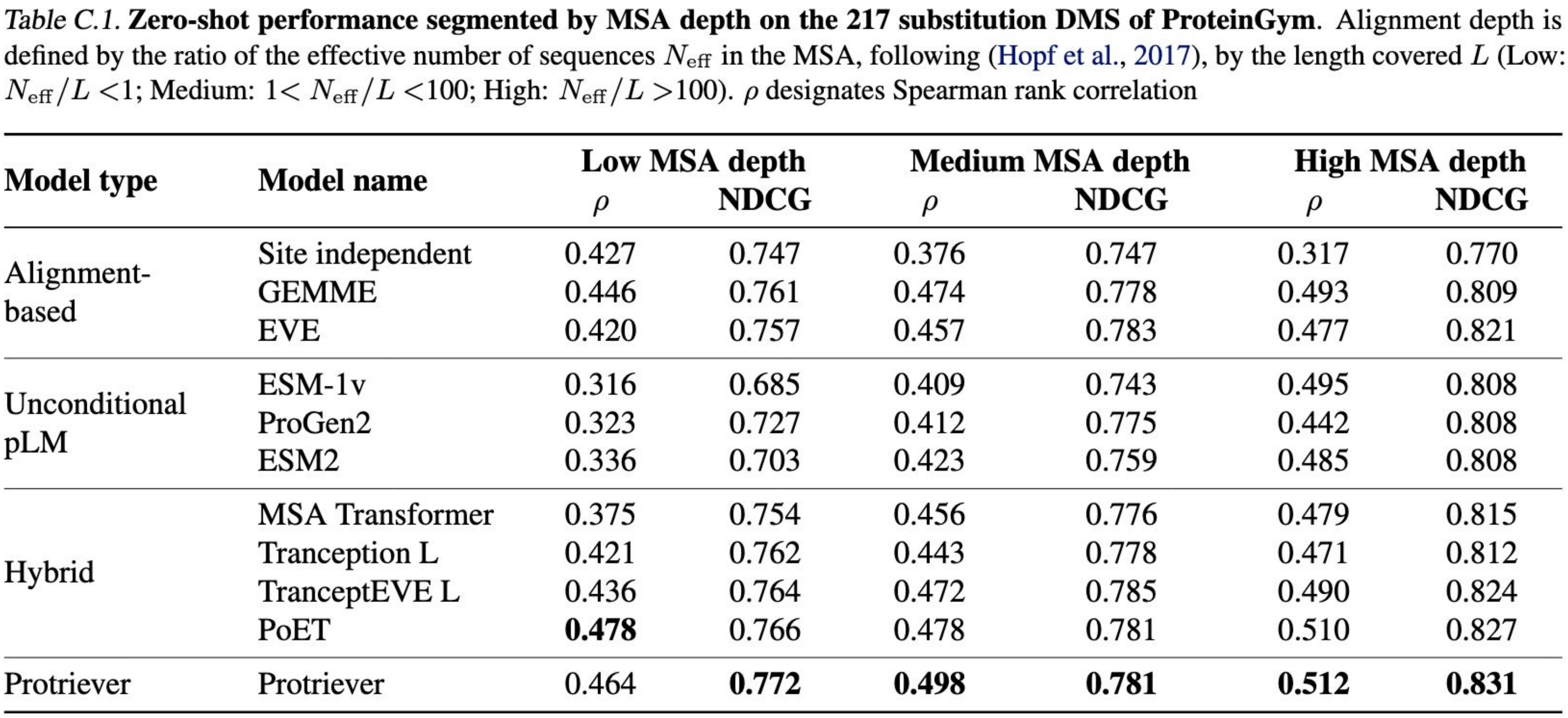


Table C.1. **Zero-shot performance segmented by MSA depth on the 217 substitution DMS of ProteinGym**. Alignment depth is defined by the ratio of the effective number of sequences N_{eff} in the MSA, following (Hopf et al., 2017), by the length covered L (Low: $N_{\text{eff}}/L < 1$; Medium: $1 < N_{\text{eff}}/L < 100$; High: $N_{\text{eff}}/L > 100$). ρ designates Spearman rank correlation

Model type	Model name	Low MSA depth		Medium MSA depth		High MSA depth	
		ρ	NDCG	ρ	NDCG	ρ	NDCG
Alignment-based	Site independent	0.427	0.747	0.376	0.747	0.317	0.770
	GEMME	0.446	0.761	0.474	0.778	0.493	0.809
	EVE	0.420	0.757	0.457	0.783	0.477	0.821
Unconditional pLM	ESM-1v	0.316	0.685	0.409	0.743	0.495	0.808
	ProGen2	0.323	0.727	0.412	0.775	0.442	0.808
	ESM2	0.336	0.703	0.423	0.759	0.485	0.808
Hybrid	MSA Transformer	0.375	0.754	0.456	0.776	0.479	0.815
	Tranception L	0.421	0.762	0.443	0.778	0.471	0.812
	TranceptEVE L	0.436	0.764	0.472	0.785	0.490	0.824
	PoET	**0.478**	0.766	0.478	0.781	0.510	0.827
Protriever	Protriever	0.464	**0.772**	**0.498**	**0.781**	**0.512**	**0.831**

Protriever performs consistently well across different evolutionary contexts, with particularly strong performance on prokaryotes and viruses

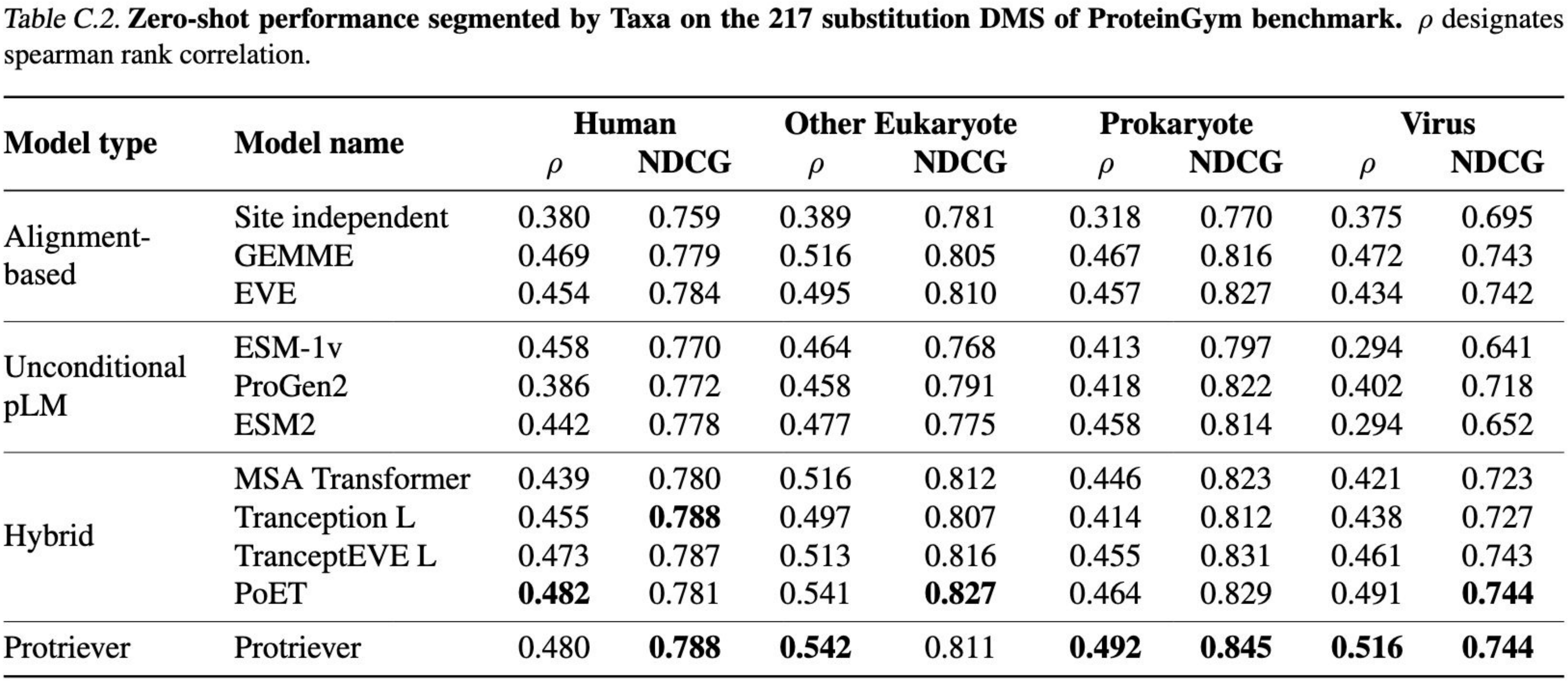
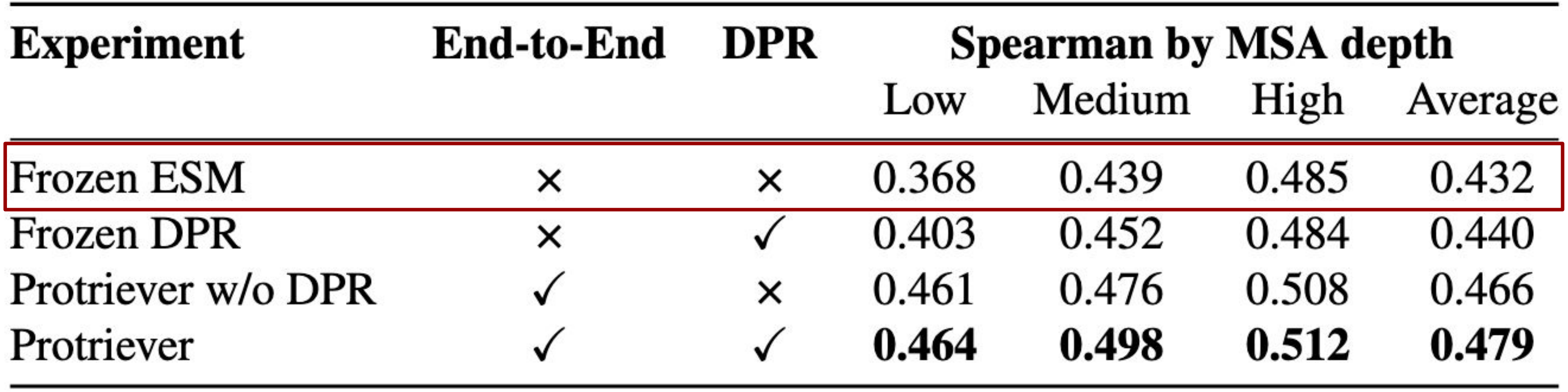


Table C.2. **Zero-shot performance segmented by Taxa on the 217 substitution DMS of ProteinGym benchmark.** ρ designates spearman rank correlation.

Model type	Model name	Human		Other Eukaryote		Prokaryote		Virus	
		ρ	NDCG	ρ	NDCG	ρ	NDCG	ρ	NDCG
Alignment-based	Site independent	0.380	0.759	0.389	0.781	0.318	0.770	0.375	0.695
	GEMME	0.469	0.779	0.516	0.805	0.467	0.816	0.472	0.743
	EVE	0.454	0.784	0.495	0.810	0.457	0.827	0.434	0.742
Unconditional pLM	ESM-1v	0.458	0.770	0.464	0.768	0.413	0.797	0.294	0.641
	ProGen2	0.386	0.772	0.458	0.791	0.418	0.822	0.402	0.718
	ESM2	0.442	0.778	0.477	0.775	0.458	0.814	0.294	0.652
Hybrid	MSA Transformer	0.439	0.780	0.516	0.812	0.446	0.823	0.421	0.723
	Tranception L	0.455	**0.788**	0.497	0.807	0.414	0.812	0.438	0.727
	TranceptEVE L	0.473	0.787	0.513	0.816	0.455	0.831	0.461	0.743
	PoET	**0.482**	0.781	0.541	**0.827**	0.464	0.829	0.491	**0.744**
Protriever	Protriever	0.480	**0.788**	**0.542**	0.811	**0.492**	**0.845**	**0.516**	**0.744**

Ablation study

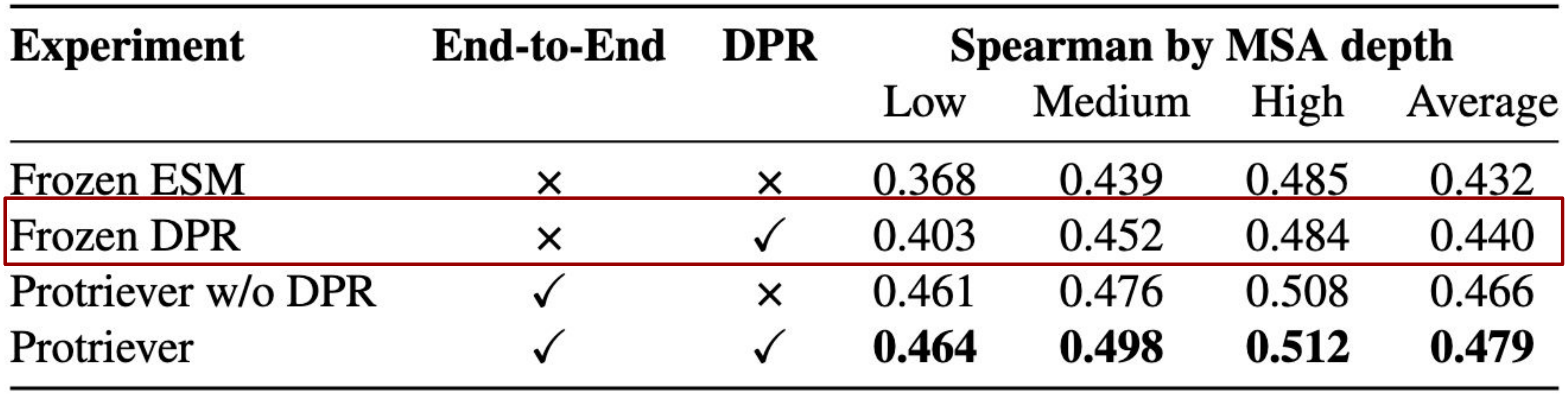
ESM capture meaningful sequence homology relationships despite not being explicitly trained for retrieval



Experiment	End-to-End	DPR	Spearman by MSA depth			
			Low	Medium	High	Average
Frozen ESM	×	×	0.368	0.439	0.485	0.432
Frozen DPR	×	✓	0.403	0.452	0.484	0.440
Protriever w/o DPR	✓	×	0.461	0.476	0.508	0.466
Protriever	✓	✓	**0.464**	**0.498**	**0.512**	**0.479**

Ablation study

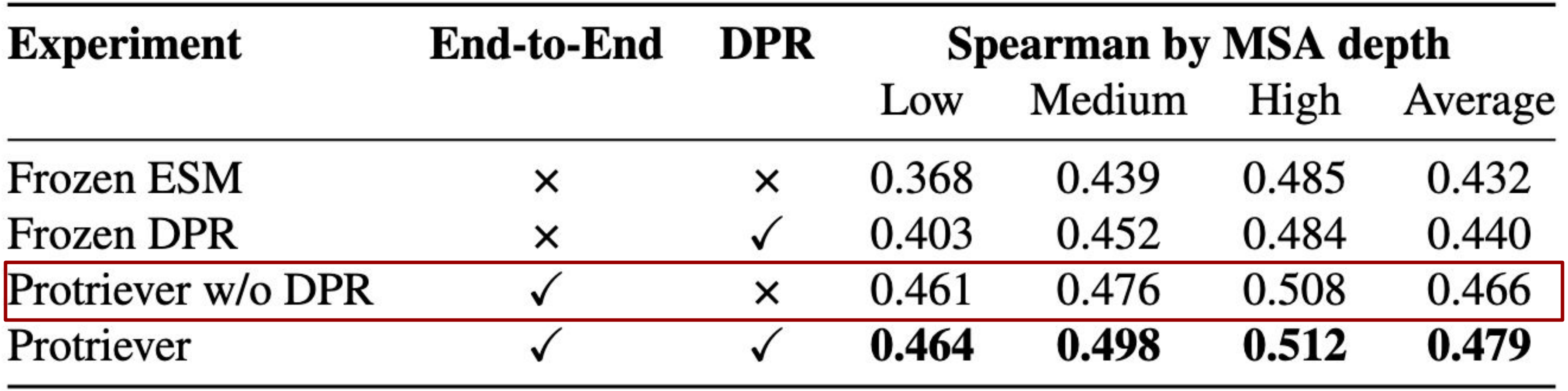
Contrastive learning for sequence retrieval is beneficial, as DPR is specifically trained on known distant sequence homologs.



Experiment	End-to-End	DPR	Spearman by MSA depth			
			Low	Medium	High	Average
Frozen ESM	×	×	0.368	0.439	0.485	0.432
Frozen DPR	×	✓	0.403	0.452	0.484	0.440
Protriever w/o DPR	✓	×	0.461	0.476	0.508	0.466
Protriever	✓	✓	**0.464**	**0.498**	**0.512**	**0.479**

Ablation study

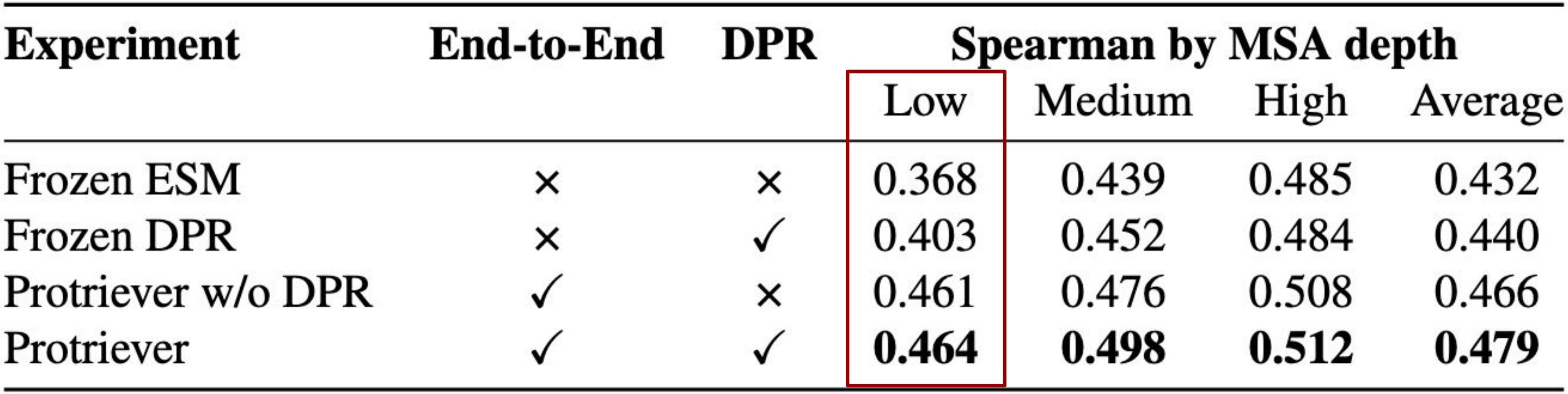
Joint optimization enables the retriever to learn representations tailored to fitness prediction, while the reader distills information about homolog usefulness for sequence reconstruction across protein families.



Experiment	End-to-End	DPR	Spearman by MSA depth			
			Low	Medium	High	Average
Frozen ESM	×	×	0.368	0.439	0.485	0.432
Frozen DPR	×	✓	0.403	0.452	0.484	0.440
Protriever w/o DPR	✓	×	0.461	0.476	0.508	0.466
Protriever	✓	✓	**0.464**	**0.498**	**0.512**	**0.479**

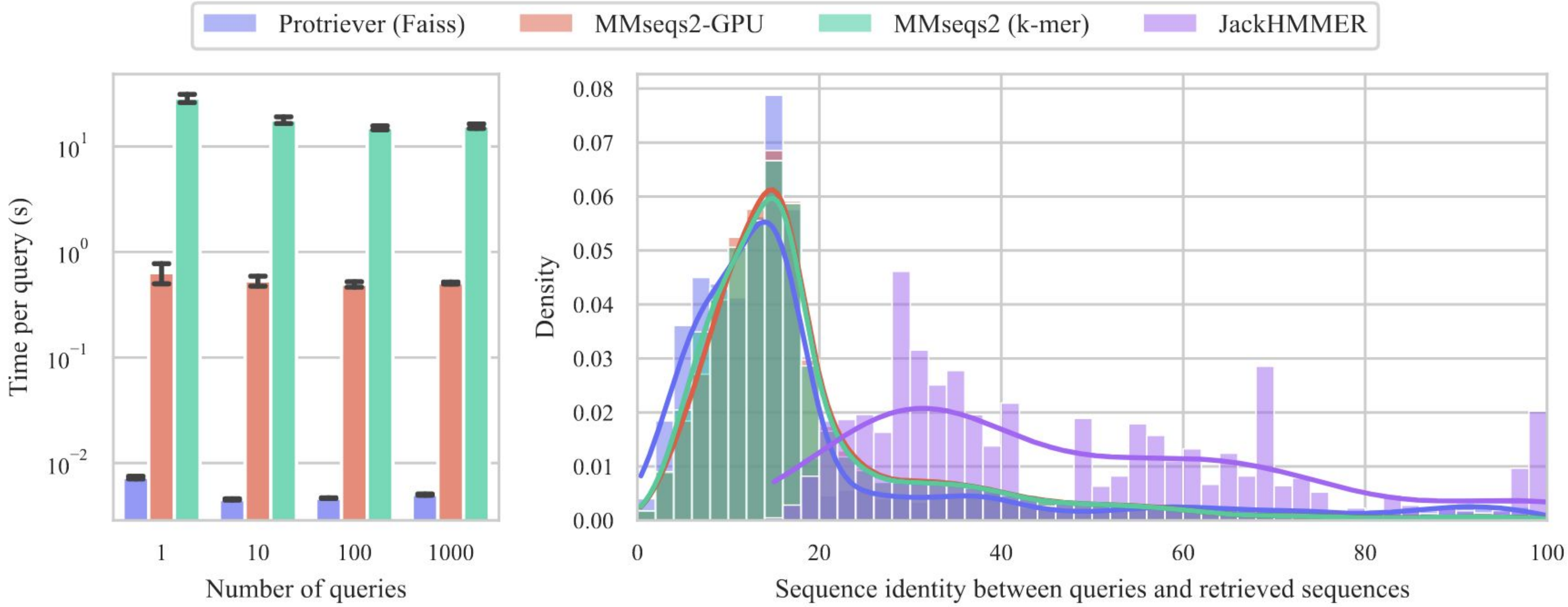
Ablation study

Largest benefits are for low-depth MSAs, where traditional alignment-based approaches struggle.

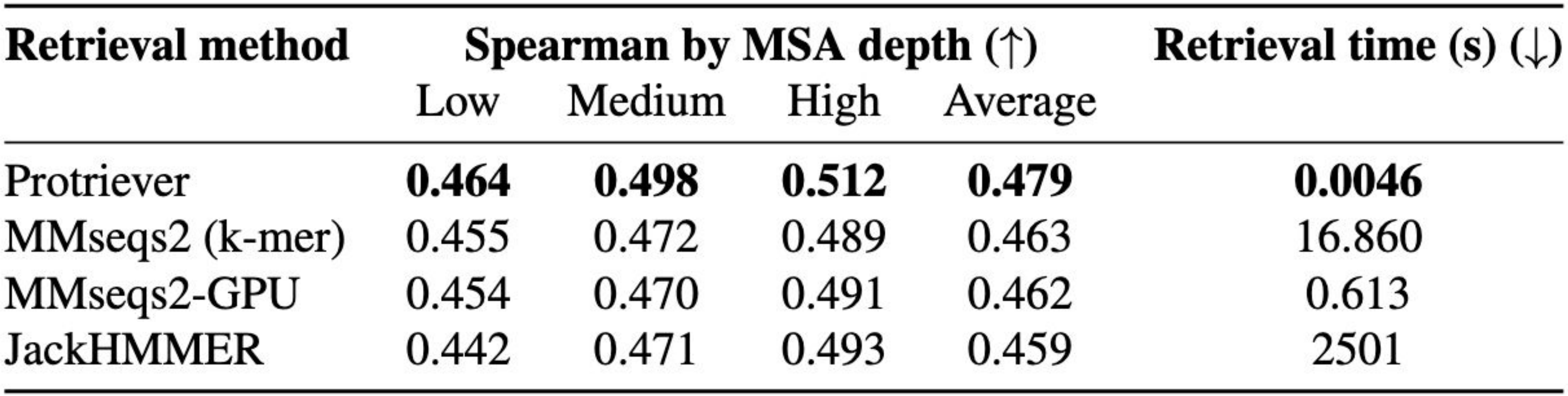


Experiment	**End-to-End**	**DPR**	**Spearman by MSA depth**			
			Low	Medium	High	Average
Frozen ESM	×	×	0.368	0.439	0.485	0.432
Frozen DPR	×	✓	0.403	0.452	0.484	0.440
Protriever w/o DPR	✓	×	0.461	0.476	0.508	0.466
Protriever	✓	✓	**0.464**	**0.498**	**0.512**	**0.479**

Protriever's vector similarity search is **at least two orders of magnitude faster** than MSA-based retrieval approaches



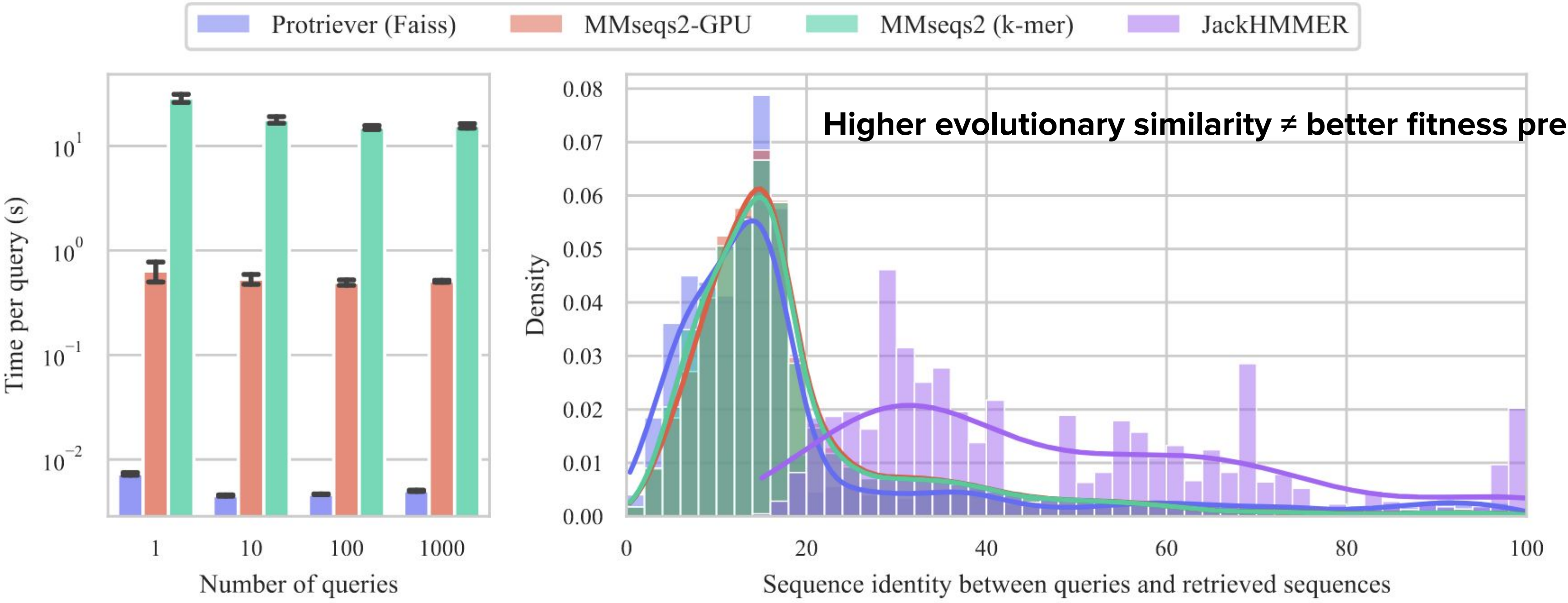
The retrieval of Protriever comes **without performance loss** and even leads to gains through the **joint retriever-reader framework.**



Retrieval method	**Spearman by MSA depth (↑)**				**Retrieval time (s) (↓)**
	Low	Medium	High	Average	
Protriever	**0.464**	**0.498**	**0.512**	**0.479**	**0.0046**
MMseqs2 (k-mer)	0.455	0.472	0.489	0.463	16.860
MMseqs2-GPU	0.454	0.470	0.491	0.462	0.613
JackHMMER	0.442	0.471	0.493	0.459	2501

Homology of retrieved sequences:

MMseqs2 and Protriever searches have considerable overlap, while JackHMMER results show substantially higher similarities

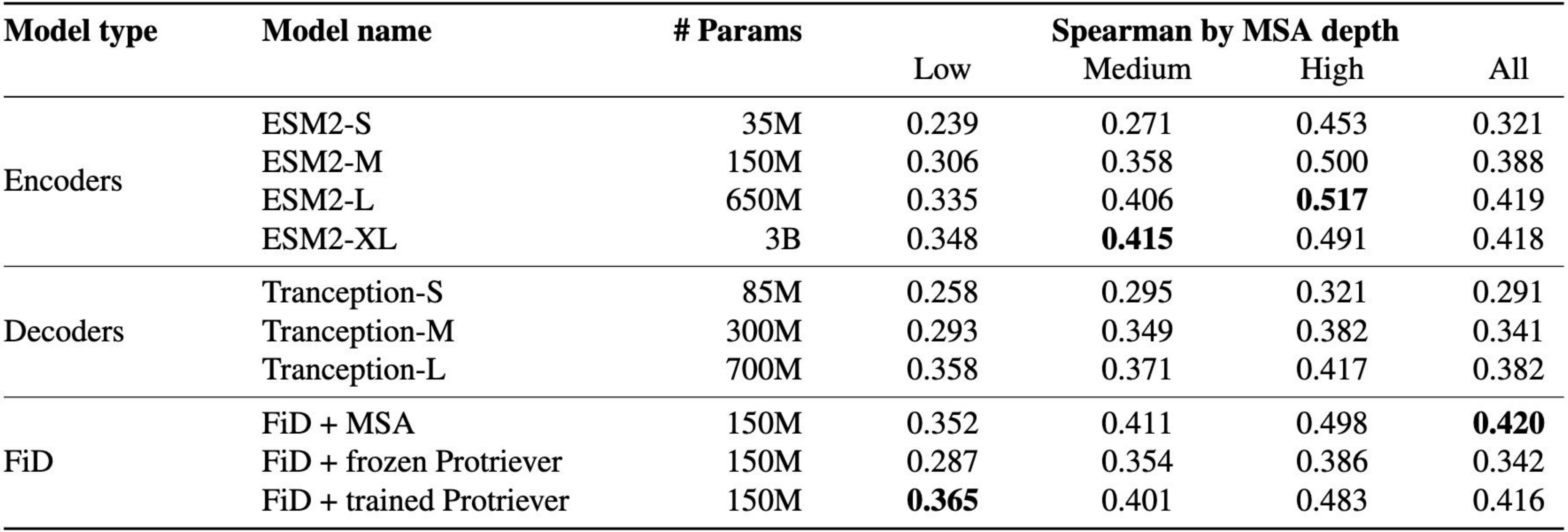


Higher evolutionary similarity ≠ better fitness prediction

Alternative architecture for the reader

Fusion in Decoder: encodes each retrieved sequence separately through an encoder, then concatenates their representations for the decoder to attend over

ESM encoder (35M) - Tranception decoder (85M), connected through cross-attention layers



Model type	**Model name**	**# Params**	**Spearman by MSA depth**			
			Low	Medium	High	All
Encoders	ESM2-S	35M	0.239	0.271	0.453	0.321
	ESM2-M	150M	0.306	0.358	0.500	0.388
	ESM2-L	650M	0.335	0.406	**0.517**	0.419
	ESM2-XL	3B	0.348	**0.415**	0.491	0.418
Decoders	Tranception-S	85M	0.258	0.295	0.321	0.291
	Tranception-M	300M	0.293	0.349	0.382	0.341
	Tranception-L	700M	0.358	0.371	0.417	0.382
FiD	FiD + MSA	150M	0.352	0.411	0.498	**0.420**
	FiD + frozen Protriever	150M	0.287	0.354	0.386	0.342
	FiD + trained Protriever	150M	**0.365**	0.401	0.483	0.416

Alternative training loss functions

Perplexity Distillation: how much each sequence improves the language model's perplexity when reconstructing the query sequence

KL-div between the retriever's relevance scores and the posterior distribution based on language model performance:

$$p_k = \frac{\exp\left(\log p_{\mathrm{LM}}\left(\mathbf{q} \mid \mathbf{d}_k\right)\right)}{\sum_{i=1}^{K} \exp\left(\log p_{\mathrm{LM}}\left(\mathbf{q} \mid \mathbf{d}_i\right)\right)}$$

Leave-One-Out Perplexity Distillation (LOOP): measures each sequence's contribution by evaluating how much the language model's reconstruction performance degrades when removing individual sequences from the retrieved set

$$p_{\mathrm{LOOP}}(\mathbf{d}_k \mid \mathbf{q}) = \frac{\exp\left(-\log p_{\mathrm{LM}}\left(\mathbf{q} \mid \mathcal{D}_K \setminus \{\mathbf{d}_k\}\right)\right)}{\sum_{i=1}^{K} \exp\left(-\log p_{\mathrm{LM}}\left(\mathbf{q} \mid \mathcal{D}_K \setminus \{\mathbf{d}_i\}\right)\right)}$$

Alternative training loss functions

EMDR performs slightly better than the PDist loss. LOOP performs slightly better than the other two, but requires many more forward passes

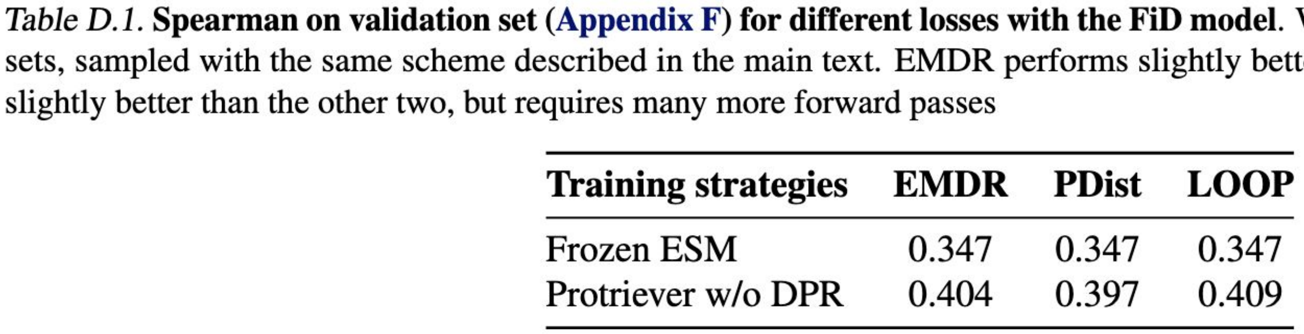


Table D.1. **Spearman on validation set (Appendix F) for different losses with the FiD model.** We evaluate the FiD model with retrieved sets, sampled with the same scheme described in the main text. EMDR performs slightly better than the PDist loss. LOOP performs slightly better than the other two, but requires many more forward passes

Training strategies	**EMDR**	**PDist**	**LOOP**
Frozen ESM	0.347	0.347	0.347
Protriever w/o DPR	0.404	0.397	0.409

Portability:

- Vector index separates encoding from retrieval → no alignments at inference
- Pre-computed index is lightweight & shareable (e.g., UniRef50 IVFPQ96×8 = 12.6GB)
- Dynamic updates: add/remove sequences without retraining

Flexibility:

- Supports domain-specific databases (e.g., GISAID, proprietary data)

Modularity:

- Compatible with diverse readers (encoder-decoder, decoder-only)
- Retriever can be replaced with structure-aware encoders
- Adaptable training objectives: autoregressive, masked LM, property/structure prediction

Conclusion

Protriever: end-to-end differentiable homology search framework

- **Performance:** state-of-the-art fitness prediction among sequence-based methods
- **Efficiency:** >100× faster than alignment-based search
- **Novelty:** joint training of retriever + reader → task-aware retrieval
- Captures functionally relevant relationships missed by alignments
- **Modular:** plug in different reader architectures & tasks
- **Interpretable:** analyze retrieved sequences

Appendix

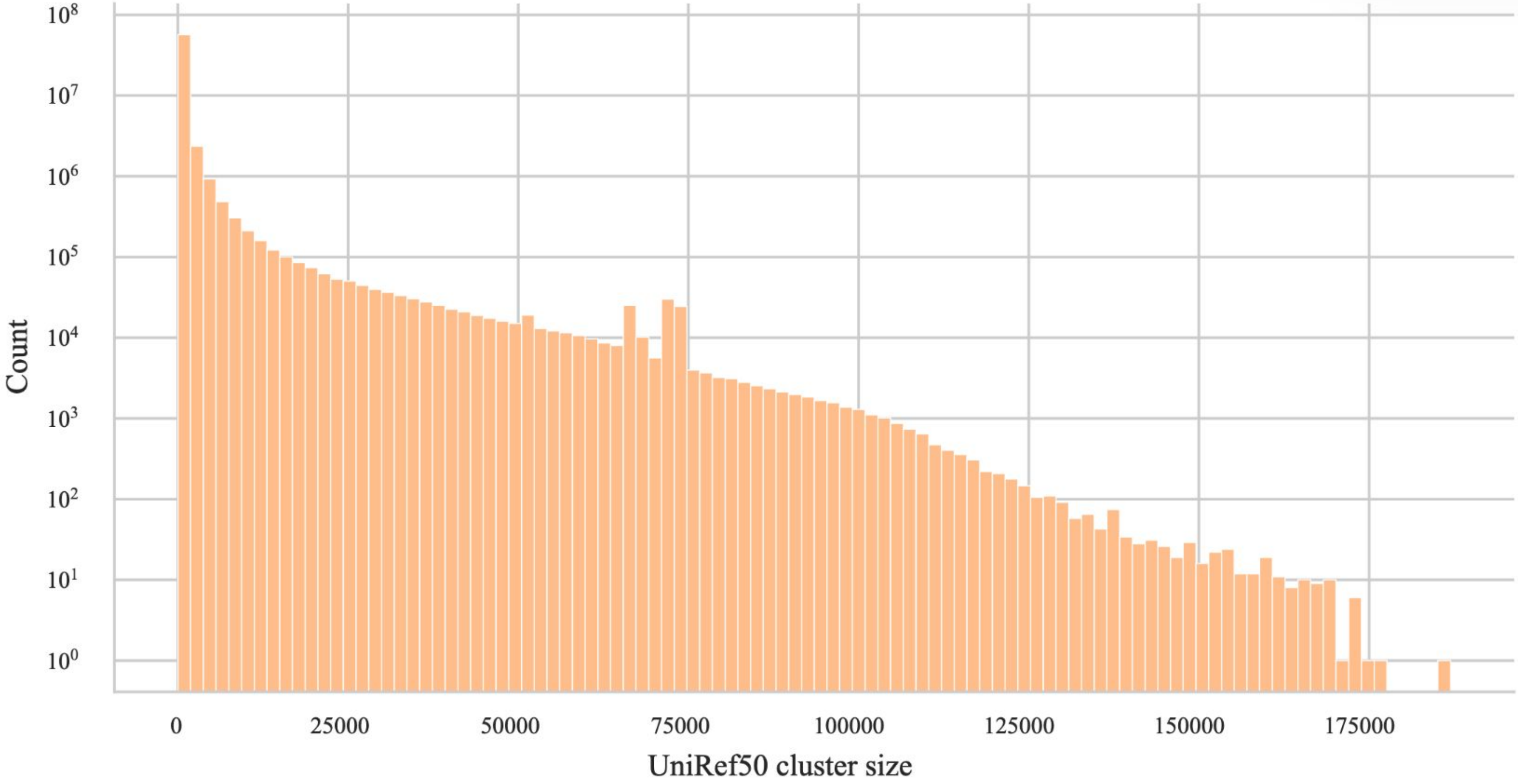


Figure G.1. **Distribution over cluster sizes of UniRef50.** Distribution of the ≈ 62 million UniRef50 clusters.

Vector Similarity

Efficient Index Search (IVF)

- **Clustering**: all entries grouped with a coarse quantizer (k-means, KIVFK_{\text{IVF}}KIVF centroids)

- **Search**:
  Query compared to all centroids
  Only the P_IVF nearest centroids (“probes”) are searched

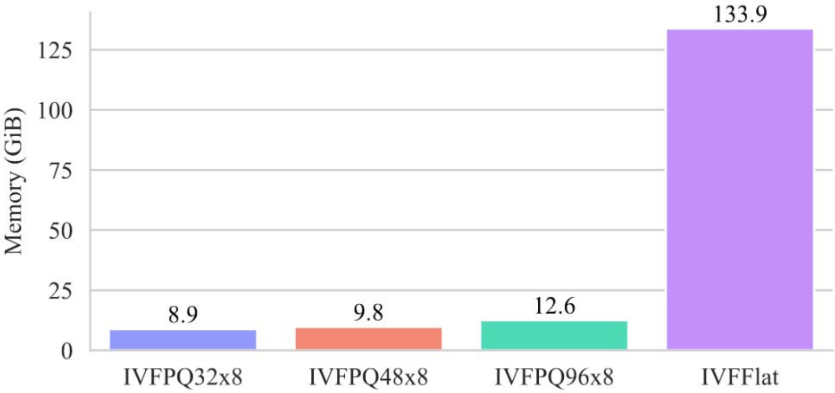
$$N_{\text{comparisons}} = K_{\text{IVF}} + P_{\text{IVF}} \frac{N}{K_{\text{IVF}}},$$

- **Efficiency**: reduces comparisons from N(full database) → much smaller subset

Vector Similarity

Product Quantization (PQ)

- Each vector split into M sub-vectors
- Each sub-vector quantized independently with k-means
- Parameters:
  M = code size (no. of sub-vectors)
  Bits = representation per sub-vector (commonly 8 or 10)



- Example: **IVFPQ32×8**
  IVF index + PQ 32 sub-vectors × 8 bits each

Choosing index parameters

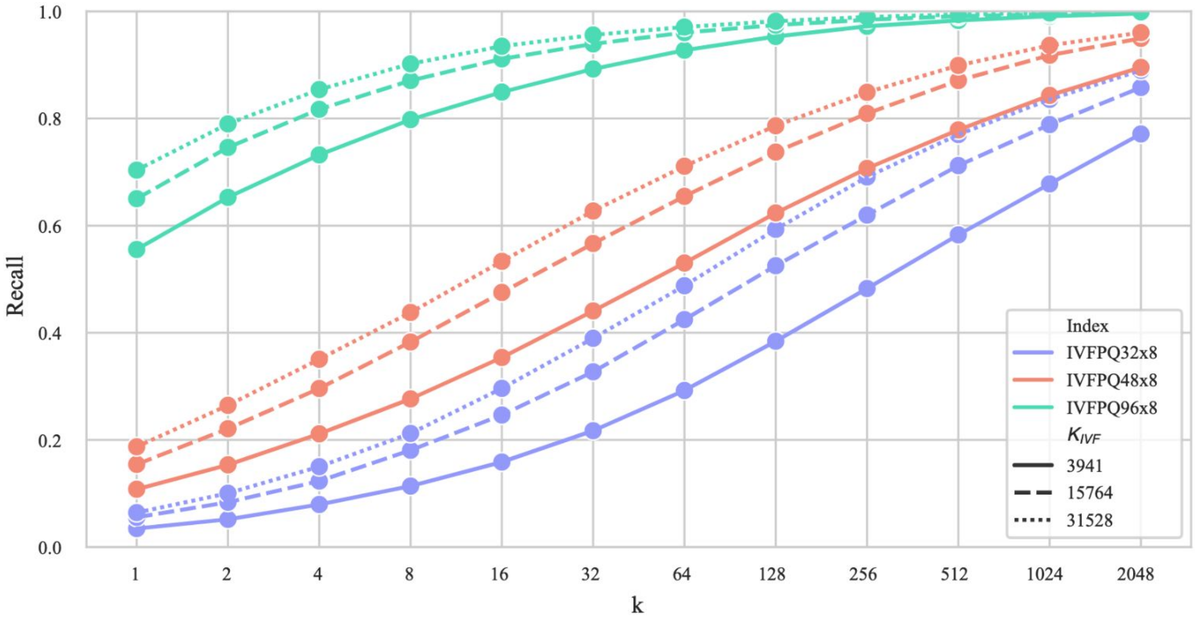
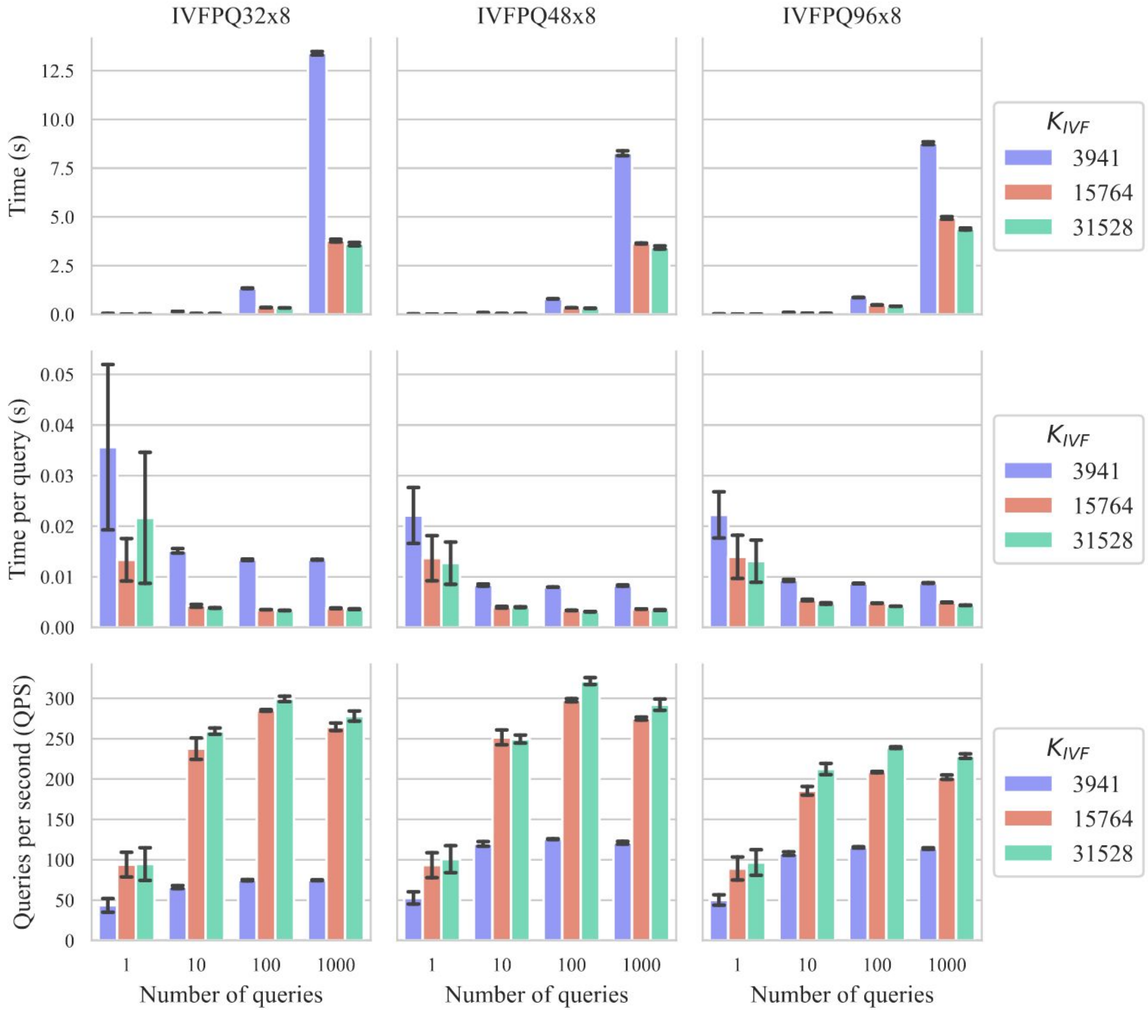


Figure A.2. Recall rate vs. neighborhood sizes for `IVFPQ` indices at different quantization levels and centroids counts. 10,000 UniRef50 sequences are randomly sampled and used as queries. For each query sequence, the 2048 nearest neighbors are found. The recall indicates whether the query sequence was successfully recovered. Decreasing the quantization from 48 sub-vectors to 96 sub-vectors leads to a significant increase in recall, while doubling the number of centroids per index from $K_{\text{IVF}} = 15764$ to $K_{\text{IVF}} = 31528$ only has a marginal performance increase.

Choosing index parameters: Search time


IVFPQ32x8
IVFPQ48x8
IVFPQ96x8
Time (s)
Time per query (s)
Queries per second (QPS)
Number of queries
K_{IVF}