Scalable embedding fusion with protein language models: insights from benchmarking text-integrated representations

Young Su Ko[1], Jonathan Parkinson[1], and Wei Wang[1,2]

1. Department of Chemistry and Biochemistry, University of California, San Diego, La Jolla, CA 92093-0359
2. Department of Cellular and Molecular Medicine, University of California, San Diego, La Jolla, CA 92093-0359

Corresponding author contact information:
Wei Wang, Ph.D.
Professor
Department of Chemistry and Biochemistry
Department of Cellular and Molecular Medicine
UCSD, 9500 Gilman Drive, La Jolla, CA 92093-0359
(858)-822-4240
wei-wang@ucsd.edu

Keywords
Protein language models, benchmarking, protein representation learning, embedding fusion

Abstract
Protein language models (pLMs) have become essential tools in computational biology, powering diverse applications from variant effect prediction to protein engineering. Central to their success is the use of pretrained embeddings–contextualized representations of amino acid sequences–which enable effective transfer learning, especially in data-scarce settings. However, recent studies have revealed that standard masked language modeling objectives often produce representations that are misaligned with the needs of downstream tasks. While scaling up model size improves performance in some cases, it does not universally yield better representations. In this study, we investigate two complementary strategies for improving pLM representations: (1) integrating text annotations through contrastive learning, and (2) combining multiple embeddings via embedding fusion. We benchmark six text-integrated pLMs (tpLMs) and three large-scale pLMs across six biologically diverse tasks, showing that no single model dominates across settings. Fusion of multiple tpLMs embeddings improves performance on most tasks but presents a computational bottleneck due to the combinatorial number of possible combinations. To overcome this, we introduce greedier forward selection, a linear-time algorithm that efficiently identifies near-optimal embedding subsets. We validate its utility through two case studies, homologous sequence recovery and protein-protein interaction prediction, demonstrating new state-of-the-art results in both. Our work highlights embedding fusion as a practical and scalable strategy for improving protein representations.

Introduction
Protein language models (pLMs) such as ESM [1] have catalyzed progress across a wide range of protein modeling tasks, including variant effect prediction [2], structure prediction [1], and protein engineering [3,4]. At the core of these advances lies the ability of pLMs to generate rich, contextualized embeddings of amino acid sequences, vector representations that encode biological features learned during pretraining. These representations have become fundamental inputs for downstream applications, powering everything from sequence classification models to structure-aware graph networks. This strategy allows researchers to offload much of the representational burden onto the pretrained model, which is especially valuable in low-data regimes. This strategy has proven effective in many settings [5–17], but recent work has revealed that the masked language modeling (MLM) objective of pLM pre-training are not always well-aligned with the representational needs of certain downstream tasks.

While scaling up pLMs might be expected to produce better, more generalizable representations, this trend does not always hold. Li et al. [18] showed that increasing ESM's size from 43M to 650M parameters improved performance on secondary structure prediction, but yielded mixed returns on tasks involving protein function and tasks where coevolutionary signals are less informative. These findings suggest that representations learned through standard MLM pre-training may be misaligned with the requirements of many biologically meaningful tasks. This raises a central question: **how can we improve protein representations to better support diverse downstream applications, especially those that do not directly benefit from scaling.**

Addressing the root cause by creating a new, well-aligned pre-training task is nontrivial. An alternative and promising direction is to leverage existing, but often ignored, protein annotations. Recent studies such as ProteinCLIP [19] and ProTrek [20] have introduced "text+protein"

language models (tpLMs) [21], which integrate textual protein annotations (e.g., UniProt functional annotations [22]) into the model's latent space. These models apply contrastive learning to align protein embeddings with corresponding text embeddings, producing representations that are often more informative than those from standalone pLMs. While these works demonstrated outperforming similar sized pLMs, such as the 650M parameter ESM2 or the 420M parameter ProtBert [23] in downstream tasks, they have not been compared to significantly larger pLMs such as ESM2 3B, Ankh, and ProtT5. Benchmarking against these scaled-up pLMs is essential for understanding whether tpLMs offer genuine representational gains, or whether their performance can be matched simply by increasing model size.

Another promising strategy for improving protein representations is to combine multiple pretrained embeddings, an ensemble learning strategy referred to as embedding fusion. Given the diversity of pLMs in architecture, scale, and training data [24], different models may capture complementary aspects of protein biology. While individual pLMs can perform well on specific tasks, no single model consistently excels across all biologically meaningful settings. In natural language processing, embedding fusion has been shown to improve performance in tasks such as hate speech detection [25–27]. In protein modeling, however, fusion has only been explored in isolated cases, such as protein function prediction [28] and post-translational modification prediction [29], and its broader utility remains unclear. We hypothesize that this underuse stems from the absence of systematic benchmarking, leaving open the question of whether embedding fusion can serve as a general-purpose strategy for improving pLM representations.

In this work, we address the central question of how protein language model (pLM) representations can be improved to better support downstream tasks. We present the results of an extensive benchmarking effort, in which we trained 1,980 deep learning models across six biologically diverse datasets. First, we benchmark six state-of-the-art (SoTA) tpLMs against three large SoTA pLMs across six diverse datasets to assess whether incorporating textual annotations yields more effective representations. Second, we explore embedding fusion as a strategy to combine the complementary strengths of pretrained models. While embedding fusion proves effective, we find that identifying the optimal combination of embeddings poses a major challenge: exhaustive search is computationally impractical.

To address this, we benchmark two classic feature selection algorithms alongside our proposed linear-time selection method for efficiently identifying high-performing embedding combinations. We find that our linear-time method offers significant speed-ups while maintaining competitive performance. Finally, we conduct case studies to demonstrate the practical utility of our approach. By applying embedding fusion with our selection algorithm to improve the representations used for two biologically meaningful tasks–protein-protein interaction prediction and homologous sequence recovery–we achieve new SoTA results in both tasks. Together, these findings introduce a practical and scalable strategy for improving protein representations, crucial for downstream tasks that are poorly aligned with current pLM pre-training.

Results and Discussion

Our study is organized into four main components:

1. **tpLMs vs. large pLMs**: We evaluate six tpLMs against three large-scale pLMs across six diverse tasks **(Fig. 1A)**.
2. **Embedding fusion**: Using tpLMs, we benchmark embedding fusion and investigate its validity as a strategy to enhance representations **(Fig. 1B)**.

3. **Embedding selection**: We assess the effectiveness of two established algorithms and our proposed linear-time algorithm for embedding selection **(Fig. 1C)**
4. **Case studies**: We apply our findings to two representative challenges, homologous sequence recovery and protein-protein interaction prediction **(Fig. 1D)**.

tpLMs: promising, but not perfect

We first curated six SoTA tpLMs: OntoProtein, ProtST, ProteinDT, ProteinCLIP, ProTrek and ESM3 **(Fig. 2A)**. The selection criteria and key characteristics of each tpLM are further described in the **Methods and Supplementary Section S1**. To evaluate their effectiveness, we compared their representations against three large-scale pLMs, ESM2 3B, Ankh, and ProtT5, across six benchmark datasets (**see Methods and Supplementary Section S2**). Following Schmirler et al. [30], we used a standardized prediction head and adopted their training regimen, including model architecture and hyperparameters.

As shown in **Fig. 2B**, based on the average performance (accuracy for the Location benchmark, Spearman's ρ for all others) on the respective test sets across five random seeds, a tpLM embedding achieves the best performance except for the AAV and GB1 benchmarks. Average performance of all models are shown in **Supplementary Table 1**. For GB1, ProTrek, the best tpLM, shares overlapping 95% confidence intervals (CIs) with ProtT5, the best model, indicating comparable performance. In the case of AAV, prior work has shown that performance continues to improve with larger model sizes, even when the task is poorly aligned with pre-training, suggesting this benchmark may benefit more from scale than representation alignment [18]. These results highlight both the promise and limitations of tpLMs: while text integration can improve representation quality, no single embedding consistently excels across all tasks. Notably, we observe the same trend for the large-scale pLMs–performance varies by task, with no pLM emerging as consistently superior. This variability led us to hypothesize that combining embeddings could combine their individual strengths to yield more robust and generalizable representations.

Embedding fusion: effective but impractical.

To investigate this hypothesis, we assessed every single possible combination of tpLM embeddings on the six benchmarks, using the same training setup as before (**Supplementary File 1**). For six tpLMs, 63 unique combinations are possible, where a combination is created by the concatenation of individual embeddings, disregarding order. While embeddings can be fused in different ways, at this stage, we establish a baseline with the most straight-forward approach. For each benchmark, we compared the embedding combination with the highest test performance with the single embedding with the highest test performance.

We observed that embedding fusion outperformed single embeddings in four out of six benchmarks, with statistically significant improvements in Spearman's rank correlation coefficient (ρ) supported by non-overlapping 95% confidence intervals (**Fig. 2C**). Although no statistically significant improvement was observed for the Stability benchmark, a numerical increase in Spearman's ρ was shown. For the Location benchmark, we observed a non-significant decrease in accuracy. Upon further investigation, ProTrek is the only tpLM to both update the text encoder and utilize the "Subcellular Location" annotations from UniProt during pre-training, suggesting its learned representations are already saturated for this task. Recognizing that embedding fusion increases the input dimension, and thus the parameter count in the projection layer, we additionally compare parameter-normalized performances **(Supplementary Table 2)** and observe the same trend of significance.

A possible concern is the increased resources necessary for embedding fusion. The best-performing combinations incurred an average 1.65× increase in runtime and 2.45× increase in peak GPU memory usage compared to the best single embedding. While these appear substantial as ratios, we emphasize that the absolute increases were modest, no more than 40 seconds in runtime or 200MB in memory. These multiplicative factors are inflated by the small size of our base model, where fixed overheads such as I/O and input projection dominate the runtime. As model size increases, the relative cost of fusion decreases, since the majority of computation shifts to deeper network layers. We demonstrate this empirically in **Supplementary Section S4**, where we show that fusion overhead does not scale linearly with model size. Thus, we reasoned these overheads are not prohibitively costly. Additionally, as Schmirler et al. [30] who ran evaluations on the same datasets with single embeddings reported, we observed differences in validation and test performance, which can be attributed to the individual dataset characteristics, such as dataset size or large domain shifts.

However, a crucial limitation is that the best combination of embeddings is dataset dependent and combining all embeddings does not lead to the best results, likely due to feature redundancy. This raises a practical challenge: how can one efficiently identify an optimal subset of embeddings without exhaustively evaluating all possibilities? Our current strategy evaluates all 63 combinations, which is computationally expensive and scales poorly as the number of available embeddings grows. To underscore the computational cost, note that ten embeddings yield 1,023 possible combinations, while fifteen create over 32,000. This combinatorial burden likely contributes to why embedding fusion remains underutilized in protein machine learning workflows, despite its clear performance benefits.

Greedy algorithms enable tractable embedding fusion.

The task of selecting an optimal set of embeddings can be framed as a feature selection problem, a well-studied challenge in classical machine learning in which the goal is to determine an optimal set of features. Greedy algorithms for feature selection such as forward selection and backward selection offer a tradeoff between performance and efficiency [31]: while they do not guarantee finding the global optimum, they can identify high-performing subsets at a fraction of the computational cost.

We adapt these strategies to the embedding fusion setting and evaluate three algorithms: forward selection, backward selection, and our proposed greedier forward selection (**Fig. 3A**). The design of greedier forward selection was motivated by two consistent trends in our benchmarking results: (1) the best-performing single embedding is nearly always included in the optimal combination, and (2) even the weakest individual embeddings occasionally contribute meaningfully to the fused representation. To balance these dynamics, greedier forward selection initializes from the top-ranked single embedding and incrementally adds others based on individual performance, evaluating combinations at each step. Full algorithmic details are provided in the **Methods**.

As shown in **Fig. 3B**, all three greedy algorithms are theoretically more efficient than exhaustive search, which scales exponentially with the number of embeddings. In **Fig. 3C**, we benchmark each algorithm across all six datasets by comparing their performance to the best test performance obtained via exhaustive search. On average, greedier forward selection achieves 99.5% of the maximum test performance identified by exhaustive search, while being 1.69× faster than forward selection (which achieved 99.8%) and 2.11× faster than backward selection

(99.2%). These results demonstrate that greedier forward selection strikes an effective balance between accuracy and efficiency, offering a scalable solution to embedding fusion that avoids combinatorial explosion.

We note that these benchmarks are designed to evaluate representation quality rather than to develop production-ready models. Our goal is to systematically test the utility of embedding fusion under controlled, reproducible conditions. Thus, as a final step in this study, we investigate how embedding fusion performs in such realistic, task-specific settings using existing state-of-the-art models.

Case Study: Strengthening protein representations with embedding fusion.

While our benchmarks were designed to systematically evaluate representation quality under controlled conditions, they do not fully capture how embedding fusion performs in real-world scenarios, where models are often fine-tuned, hyperparameter-optimized, or enhanced with techniques like LoRA. To assess whether the benefits of fusion persist in these practical settings, we conduct two case studies on biologically meaningful tasks: homologous sequence recovery and protein-protein interaction (PPI) prediction [19].

In each case, we apply embedding fusion–guided by greedier forward selection–to augment an existing state-of-the-art model. We fix all other components of the pipeline to isolate the contribution of improved representations. For the PPI task, we additionally compare against previously reported results from fine-tuning and LoRA-based approaches to contextualize our approach within the broader landscape of machine-learning strategies. Finally, we evaluate how close our heuristic selection gets to the global optimum by running exhaustive search. These case studies move beyond controlled benchmarking to demonstrate how fusion can enhance the performance of real, task-specific models, reinforcing its utility as a practical and accessible strategy.

Embedding fusion enabled more accurate homologous sequence recovery.

The homologous sequence recovery task evaluates protein embeddings in a nearest-neighbor retrieval setting, where predictions are made by comparing embeddings directly via cosine similarity. More specifically, a query protein embedding is compared against a database of embeddings using cosine similarity and retrieves the most similar embedding. A retrieval is considered correct if retrieved protein shares the same CATH [32] superfamily **(Fig. 4A)**. Unlike our deep learning benchmarks, this task requires no training or parameter tuning, offering a complementary perspective on representation quality. Additionally, the reduced computational burden enables evaluation across all pairwise combinations of the six tpLMs and three large pLMs.

The previous state-of-the-art was introduced by Wu et al., who used ProteinCLIP embeddings to achieve higher top-1 retrieval accuracy than MMseqs2 [33]. Using greedier forward selection, we identified a fused embedding composed of Ankh, ProtT5, ESM3, ProteinCLIP, and ProTrek that achieves an accuracy of 83.1%, surpassing the previous best of 80.8% **(Fig. 4B)**. Notably, this combination was also identified as the global optimum when compared against all possible combinations via exhaustive search.

To better understand how embedding fusion alters the structure of the representation space, we visualized all CATH sequences using t-SNE, comparing our fused embeddings **(Fig. 4C)** to the previously best-performing ProteinCLIP embeddings **(Fig. 4D)**. We used 3,000 iterations, cosine

distance, and a perplexity of 50. The fused embeddings exhibit improved class separation and tighter intra-class clustering, suggesting that they better capture the distinct structural features of each CATH class. While t-SNE provides qualitative insight, we also sought a quantitative measure of improvement. We therefore computed per-class retrieval accuracies and observed consistent gains across all classes **(Fig. 4E)**, in agreement with the visual patterns.

These results demonstrate that embedding fusion, when guided by a lightweight selection strategy, can meaningfully enhance protein representations even in a training-free retrieval setting. The improvement over the previous state-of-the-art, combined with the ability of greedier forward selection to recover the global optimum, underscores the practical viability of fusion in real-world applications. Moreover, the qualitative and quantitative gains in class separation and retrieval accuracy suggest that fused embeddings capture richer and more discriminative structural features than any individual model alone. Notably, the best-performing combination includes both Ankh and ProtT5, illustrating that the benefits of embedding fusion extend beyond text-integrated models and generalize to text-free pLMs as well.

A new frontier for sequence-based protein-protein interaction prediction.

Our second case study returns to a deep learning setting, focusing on the widely studied challenge of sequence-based PPI prediction. Such models are essential for high-throughput in silico screening, rational binder design, and the discovery of novel molecular interactions. However, early approaches were often confounded by data leakage, where high sequence similarity between training and test sets inflated performance metrics [34]. To address this, Bernett et al. introduced a rigorously filtered benchmark of human PPIs that minimizes sequence redundancy and has since become the gold standard for evaluating generalizability in this task.

To examine how embedding fusion performs in this real-world, leakage-free setting, we compile results from recent literature, including models that leverage fine-tuning and LoRA on ESM2 [19,35,36]. We adopt the architecture and hyperparameters of NaderiAlizadeh and Singh [35], who report the highest accuracy on the Bernett benchmark (**Fig. 4F**; see **Methods** for full details). We modify only the embedding module, using greedier forward selection to identify the most effective combination of pretrained embeddings based on validation performance. This setup allows us to directly assess whether improved representations alone can enhance performance in an already-optimized model.

The selected combination, ProteinCLIP, ProtST, and ProTrek, achieved the highest average AUPRC and outperformed all individual tpLM embeddings **(Supplementary Table 3)**. On the test set, this combination achieves a new state-of-the-art in both AUPRC and accuracy (**Fig. 4G**). Exhaustive search revealed that the true best-performing combination includes an additional model, ProteinDT, further improving test AUPRC and pushing performance beyond previously reported baselines. Importantly, both the selected and global-optimum combinations outperform any single tpLM, demonstrating the consistent benefit of fusion in this task.

To better understand the practical implications of this improvement, we simulated a realistic use case in which a classifier is used to select top-ranked interaction candidates for experimental validation. We re-implemented the NaderiAlizadeh model, including its original use of ESM2 650M embeddings (see **Supplementary Section S5** for reproduction details) and compared it against the fusion-enhanced model. For each model, we identified the top-k predictions (k = 10,

100, 1000) from the test set with the highest predicted probability of interaction and calculated the precision and number of false positives at each threshold **(Fig. 4H)**.

Both models achieved perfect precision in the top-10 predictions. However, at broader thresholds, the fusion-based model demonstrated markedly better performance, achieving higher precision and fewer false positives and producing less than half as many false positives as the baseline. These findings highlight how improved representations from fusion can yield more robust classifiers, better able to generalize and prioritize meaningful interactions, especially in scenarios where false positives are costly.

Concluding Remarks

This study set out to investigate how pLM representations can be improved, particularly for tasks that remain misaligned with current pre-training objectives. We first benchmarked six tpLMs against three large-scale pLMs across six diverse tasks. While tpLMs frequently outperformed larger models, no single model consistently yielded the best representations across all benchmarks. These results underscore a key limitation in current approaches: relying on a single model, regardless of size or supervision, does not capture the full spectrum of biologically relevant features.

To address this, we explored embedding fusion as a general-purpose strategy for combining complementary representations. While fusion consistently improved downstream performance, its adoption has been limited by the computational burden of exhaustively evaluating all embedding combinations. To make embedding fusion tractable, we benchmarked multiple selection algorithms and introduced greedier forward selection, a linear-time algorithm that identifies near-optimal combinations with significantly reduced cost. This method enabled practical application of fusion without sacrificing performance

We then tested the real-world utility of this approach through two case studies: homologous sequence recovery and PPI prediction. In both settings, embedding fusion paired with greedier forward selection led to new SoTA results, outperforming previous baselines. These findings show that improved representations can substantially enhance model performance and practical utility. More broadly, they demonstrate how our approach transforms embedding fusion from a promising but computationally intractable idea into a usable tool for protein machine learning.

We also acknowledge the limitations of our study. Recent works have highlighted the species bias present in the training data for pLMs as well as the potential data leakage between the pre-training data and the downstream task [37,38]. As a result, performance with any pLM embeddings on benchmark datasets may be over optimistic in regard to its ability to generalize to unseen or underrepresented proteins. While datasets used in this study have been partitioned to consider the sequence similarity between the training and test splits, the pre-training data and test data are still relatively small and limited. As foundation models continue to scale and absorb increasingly diverse training data, future work should focus on evaluating representation robustness and transferability under more challenging generalization settings.

Methods

Protein language models

Table 1 provides an overview of selected tpLMs. We describe the key features of each tpLM, below. additional details, including the composition of pre-training data, are described in **Supplementary Section S1**.

OntoProtein utilized Gene Ontology and Gene Annotation terms during pre-training and showed improvements on the TAPE benchmarks [39], protein-protein interaction prediction, and protein function prediction [40]. ProtST performed contrastive learning between PubMedBERT embeddings of Swiss-Prot text annotations and pLM embeddings to achieve further improvements in downstream performance [41]. Similarly, ProteinDT introduced the Contrastive Language and Protein pre-training (ProteinCLAP) approach to align the representations between SciBert [42] embeddings of Swiss-Prot annotations and ProtBert [23] embeddings. ProteinCLIP used contrastive learning to align protein sequence representations from pre-trained pLMs to UniProt function descriptions embedded by OpenAI's text embedding model [19]. The tri-modal pLM ProTrek was trained with a contrastive learning objective between sequence, structure, and function, aligning GPT-4-summarized UniProt descriptions embedded by PubMedBERT and protein sequence embedded by ESM2 [20]. Lastly, ESM3 takes a different tri-modal approach, representing sequence, structure, and function annotations as discrete tokens and predicting the masked tokens across the three different modalities as the pre-training task [43].

Datasets

We adopt the same six benchmarks used by Schmirler et al.'s study on fine-tuning pLMs [30]. We exclude the disorder [44] and secondary structure prediction [45,46] tasks because protein structure tasks have already been shown to be well aligned with current pLM pre-training [18]. The datasets for homologous sequence recovery PPI prediction have previously been used by Wu et al. [19]. Detailed descriptions of datasets as well as dataset statistics are described in **Supplementary Section S2 and Supplementary Table 4**.

Training and Implementation

All models used in this study are re-implementations of existing architectures [19,30,35], which are described below. Models are implemented in PyTorch v2.2.2 and trained on one NVIDIA RTX A6000 GPU with 48Gb memory. All deep-learning models are run five times using five different seeds (2,4,8,16, and 32), checkpointed based on the best validation performance, and the average performance on the test set across seeds is reported. Technical details regarding embedding generation, implementation of embedding fusion, and the metrics used in this study are described in **Supplementary Section S6 and S7.**

Evaluation Metrics

For benchmarking tpLMs and embedding fusion, we measure two metrics:

Spearman's ρ (Spearman's rank correlation coefficient) measures the monotonic relationship between the predicted and true values by comparing their rank orderings.

Top-1 accuracy is used for classification tasks where a single label is predicted for each input. It reflects the proportion of correct predictions among all samples and is used for the Location benchmark.

For the homologous sequence recovery task, we also use top-1 accuracy, where a retrieval is considered correct if the nearest-neighbor sequence (based on cosine similarity in embedding space) belongs to the same homologous superfamily as the query.
For the protein-protein interaction (PPI) prediction task, we report several standard metrics for binary classification:
Accuracy: The fraction of correct predictions across all positive and negative samples.
AUPRC (Area Under the Precision-Recall Curve): Measures the trade-off between precision and recall across thresholds. It is especially informative under class imbalance, as it focuses on the performance for the positive class.
F1 Score: The harmonic mean of precision and recall, providing a balanced metric when false positives and false negatives carry similar costs.
Matthews Correlation Coefficient (MCC): A balanced metric that accounts for all four values of the confusion matrix. It is considered a robust overall measure of binary classifier performance, particularly in imbalanced settings.

Regressors and Classifier for the six benchmark datasets

The use of a shallow multilayer perceptron (MLP), referred to as the prediction head, is standard practice in recent pLM benchmarking studies, such as PEER [47], which use prediction heads to isolate the representational power of the pLM. This design allows for fair comparison across models and facilitates cross-referencing of results across studies.

We adopt the prediction head used by Schmirler et al. [30]. All models begin with batch normalization of the input data, followed by a hidden layer, ReLU activation, dropout, and an output layer. The output dimension is 10 for the subcellular location classifier and 1 for all other regressors. The loss function is Cross Entropy Loss for the subcellular location classifier and Mean Squared Error Loss for all other regressors. The loss is optimized via Adam without a learning rate scheduler. The hidden dimension size is set to 32 and the dropout rate is set to 20%. While all other hyperparameters are unchanged from Schmirler et al., we scaled the batch size from 8 to 4096 to reduce training time. We scaled the original learning rate from 0.0001 to $0.0001\times\sqrt{512}$ based on the guidance by Malladi et al. [48] to match the change in batch size. All hyperparameters are displayed in **Supplementary Table 5**.

Embedding Selection Algorithms

Forward Selection

Forward selection begins with an empty set and iteratively adds the embedding that most improves performance when combined with the current set. At each step, all remaining embeddings are evaluated in combination with the current subset, and the one that yields the best validation performance is added. This process continues until no further improvement is observed or all embeddings have been evaluated. This algorithm is $O(N^2)$ in the worst case, due to repeated evaluation of all remaining candidates at each step.
Algorithm pseudocode:

1. Initialize selected set $S = \emptyset$
2. While improvement is observed:
    - For each $e \in S$, evaluate performance of $S \cup \{e\}$
    - Add e^* that gives the highest performance
3. Return best-performing subset S

Backward Selection

Backward selection starts with the full set of embeddings and iteratively removes the embedding whose removal most improves (or least degrades) performance. This algorithm is also $O(N^2)$, as each step evaluates all remaining removals.

Algorithm pseudocode:

1. Initialize selected set $S = \{all\ embeddings\}$
2. While performance does not degrade:
    - For each $e \in S$, evaluate performance of $S \setminus \{e\}$
    - Remove e^* that yields the highest performance
3. Return best-performing subset S

Greedier Forward Selection

We begin by evaluating the performance of each individual embedding on the validation set and ranking them from best to worst. Let the ranked list be $[e_1,e_2,\ldots,e_N]$, where e_1 yields the highest standalone performance. Starting with the top-performing embedding $\{e_1\}$, we iteratively evaluate the effect of adding the next-best ranked embedding to the current set. That is, we test $\{e_1,e_2\}$, then $\{e_1,e_2,e_3\}$, and so on. At each step, an embedding is added only if it improves performance on the validation set; otherwise, it is skipped. This process continues until all embeddings have been considered. This method requires N evaluations for single-embedding ranking and up to $N-1$ additional evaluations for combinations, making it significantly more efficient than standard forward or backward selection. See **Supplementary Figure 1** for pseudocode.

Homologous sequence recovery

We use the state-of-the-art homologous sequence recovery framework from Wu et al. [19]. Given a list of query proteins and a protein database, we calculate the cosine similarity between the query protein embedding and embedding of every protein in the database. We retrieve the protein with the highest similarity to the query, excluding itself, and check if the query protein and the retrieved protein belong to the same CATH superfamily. If they belong to the same homologous superfamily, we consider that a successful retrieval. The reported accuracy is the number of successful retrievals divided by the number of queries made. There are no hyperparameters for this retrieval pipeline. Following Wu et al., we L2-normalize all embeddings before the retrieval pipeline.

Protein-protein interaction classifier

We use the state-of-the-art protein-protein interaction classifier used by NaderiAlizadeh and Singh [35]. The protein-protein interaction classifier uses one projection layer followed by a ReLU activation. The binary prediction (0 or 1) is determined by rounding the cosine similarity between the two protein projections at the 0.5 threshold. The model uses Binary Cross Entropy as the loss function, which is optimized using the AdamW optimizer paired with a Cosine Annealing Warm Restarts learning rate scheduler that restarts every 10 epochs. The hidden dimension size is set to 1024. The weights from the epoch with the highest validation AUPRC is used on the test set. These hyperparameters are the set by NaderiAlizadeh and Singh [35]. While we scaled the batch size from 32 to 4096, we kept the learning rate constant as no change in performance was observed. Exact hyperparameters are shown in **Supplementary Table 6**.

Key Points

- No single protein language model representation consistently performs best across all biological tasks, highlighting a need for methods that can combine complementary strengths of different embeddings.
- Embedding fusion improves the quality of protein representations but the best combination is task-specific and hard to identify without the costly exhaustive search.
- We propose greedier forward selection, a fast and scalable algorithm that enables near-optimal embedding fusion, leading to new state-of-the-art results in homologous sequence recovery and protein-protein interaction prediction.

Data and Code Availability

The data and code necessary to reproduce the results from this study are available at https://github.com/Wang-lab-UCSD/Benchmarking-tpLMs.

Funding

This work was partially supported by the National Institutes of Health [R21AI158114, R01AI150282].

Author Contributions

Young Su Ko: methodology, software, investigation, and writing (original draft). **Jonathan Parkinson:** software, validation, writing (review and editing). **Wei Wang:** supervision and writing (review and editing).

Ethics declarations

The authors declare no competing interests.

Author Information

Young Su Ko is a PhD candidate at the University of California, San Diego.

Jonathan Parkinson is a Project Scientist at the University of California, San Diego.

Wei Wang is a Professor at the University of California, San Diego.

References

1. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379,** 1123–1130 (2023).
2. Brandes, N., Goldman, G., Wang, C. H., Ye, C. J. & Ntranos, V. Genome-wide prediction of disease variant effects with a deep protein language model. *Nat Genet* **55,** 1512–1522 (2023).
3. Hie, B. L. *et al.* Efficient evolution of human antibodies from general protein language models. *Nat Biotechnol* **42,** 275–283 (2024).
4. Madani, A. *et al.* Large language models generate functional protein sequences across diverse families. *Nat Biotechnol* **41,** 1099–1106 (2023).
5. Singh, R., Sledzieski, S., Bryson, B., Cowen, L. & Berger, B. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proc. Natl. Acad. Sci. U.S.A.* **120,** e2220778120 (2023).
6. Roche, R., Moussad, B., Shuvo, M. H., Tarafder, S. & Bhattacharya, D. EquiPNAS: improved protein–nucleic acid binding site prediction using protein-language-model-informed equivariant deep graph neural networks. *Nucleic Acids Research* **52,** e27–e27 (2024).
7. Shenoy, A., Kalakoti, Y., Sundar, D. & Elofsson, A. M-Ionic: prediction of metal-ion-binding sites from sequence using residue embeddings. *Bioinformatics* **40,** btad782 (2024).
8. Chen, T., Zhang, Y. & Chatterjee, P. moPPIt: *De Novo* Generation of Motif-Specific Binders with Protein Language Models. Preprint at https://doi.org/10.1101/2024.07.31.606098 (2024)
9. Du, Z. *et al.* pLM4ACE: A protein language model based predictor for antihypertensive peptide screening. *Food Chemistry* **431,** 137162 (2024).
10. Jahn, L. R., Marquet, C., Heinzinger, M. & Rost, B. Protein embeddings predict binding residues in disordered regions. *Sci Rep* **14,** 13566 (2024).
11. Palacios, A. V. *et al.* SumoPred-PLM: human SUMOylation and SUMO2/3 sites Prediction using Pre-trained Protein Language Model. *NAR Genomics and Bioinformatics* **6,** lqae011 (2024).
12. Yadav, S., Vora, D. S., Sundar, D. & Dhanjal, J. K. TCR-ESM: Employing protein language embeddings to predict TCR-peptide-MHC binding. *Computational and Structural Biotechnology Journal* **23,** 165–173 (2024).
13. Alvarez, J. A. E. & Dean, S. N. TEMPRO: nanobody melting temperature estimation model using protein embeddings. *Sci Rep* **14,** 19074 (2024).
14. Pudžiuvelytė, I. *et al.* TemStaPro: protein thermostability prediction using sequence representations from protein language models. *Bioinformatics* **40,** btae157 (2024).
15. Zhu, Y.-H., Liu, Z., Liu, Y., Ji, Z. & Yu, D.-J. ULDNA: integrating unsupervised multi-source language models with LSTM-attention network for high-accuracy protein–DNA binding site prediction. *Briefings in Bioinformatics* **25,** bbae040 (2024).
16. Savojardo, C., Manfredi, M., Martelli, P. L. & Casadio, R. DDGemb: predicting protein stability change upon single- and multi-point variations with embeddings and deep learning. *Bioinformatics* btaf019 (2025). doi:10.1093/bioinformatics/btaf019
17. Ko, Y. S., Parkinson, J., Liu, C. & Wang, W. TUnA: an uncertainty-aware transformer model for sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics* **25,** bbae359 (2024).
18. Li, F.-Z., Amini, A. P., Yue, Y., Yang, K. K. & Lu, A. X. Feature Reuse and Scaling: Understanding Transfer Learning with Protein Language Models. Preprint at https://doi.org/10.1101/2024.02.05.578959 (2024)
19. Wu, K. E., Chang, H. & Zou, J. ProteinCLIP: enhancing protein language models with

natural language. Preprint at https://doi.org/10.1101/2024.05.14.594226 (2024)
20. Su, J., Zhou, X., Zhang, X. & Yuan, F. ProTrek: Navigating the Protein Universe through Tri-Modal Contrastive Learning. Preprint at https://doi.org/10.1101/2024.05.30.596740 (2024)
21. Pei, Q. *et al.* Leveraging Biomolecule and Natural Language through Multi-Modal Learning: A Survey. Preprint at http://arxiv.org/abs/2403.01528 (2024)
22. The UniProt Consortium *et al.* UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* **51,** D523–D531 (2023).
23. A. Elnaggar *et al.* ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44,** 7112–7127 (2022).
24. Hu, B. *et al.* Protein Language Models and Structure Prediction: Connection and Progression. Preprint at http://arxiv.org/abs/2211.16742 (2022)
25. Badri, N., Kboubi, F. & Chaibi, A. H. Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection. *Procedia Computer Science* **207,** 769–778 (2022).
26. Khan, M. A., Yadav, N., Jain, M. & Goyal, S. The Art of Embedding Fusion: Optimizing Hate Speech Detection. Preprint at http://arxiv.org/abs/2306.14939 (2023)
27. Lester, B., Pressel, D., Hemmeter, A., Choudhury, S. R. & Bangalore, S. Multiple Word Embeddings for Increased Diversity of Representation. Preprint at http://arxiv.org/abs/2009.14394 (2020)
28. Chua, Z. M., Rajesh, A., Sinha, S. & Adams, P. D. PROTGOAT□: Improved automated protein function predictions using Protein Language Models. Preprint at https://doi.org/10.1101/2024.04.01.587572 (2024)
29. Pokharel, S., Pratyush, P., Ismail, H. D., Ma, J. & Kc, D. B. Integrating Embeddings from Multiple Protein Language Models to Improve Protein O-GlcNAc Site Prediction. *Int J Mol Sci* **24,** 16000 (2023).
30. Schmirler, R., Heinzinger, M. & Rost, B. Fine-tuning protein language models boosts predictions across diverse tasks. Preprint at https://doi.org/10.1101/2023.12.13.571462 (2023)
31. Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3,** 1157–1182 (2003).
32. Sillitoe, I. *et al.* CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Research* **47,** D280–D284 (2019).
33. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* **35,** 1026–1028 (2017).
34. Bernett, J., Blumenthal, D. B. & List, M. Cracking the black box of deep sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics* **25,** bbae076 (2024).
35. NaderiAlizadeh, N. & Singh, R. Aggregating Residue-Level Protein Language Model Embeddings with Optimal Transport. Preprint at https://doi.org/10.1101/2024.01.29.577794 (2024)
36. Sledzieski, S. *et al.* Democratizing protein language models with parameter-efficient fine-tuning. *Proceedings of the National Academy of Sciences* **121,** e2405840121 (2024).
37. Ding, F. & Steinhardt, J. Protein language models are biased by unequal sequence sampling across the tree of life. Preprint at https://doi.org/10.1101/2024.03.07.584001 (2024)
38. Hermann, L., Fiedler, T., Nguyen, H. A., Nowicka, M. & Bartoszewicz, J. M. Beware of Data Leakage from Protein LLM Pretraining. Preprint at https://doi.org/10.1101/2024.07.23.604678 (2024)
39. Rao, R. *et al.* Evaluating Protein Transfer Learning with TAPE. *Adv Neural Inf Process Syst* **32,** 9689–9701 (2019).

40. Zhang, N. *et al.* OntoProtein: Protein Pretraining With Gene Ontology Embedding. Preprint at http://arxiv.org/abs/2201.11147 (2022)
41. Xu, M., Yuan, X., Miret, S. & Tang, J. ProtST: Multi-Modality Learning of Protein Sequences and Biomedical Texts. Preprint at http://arxiv.org/abs/2301.12040 (2023)
42. Beltagy, I., Lo, K. & Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. Preprint at http://arxiv.org/abs/1903.10676 (2019)
43. Hayes, T. *et al.* Simulating 500 million years of evolution with a language model. Preprint at https://doi.org/10.1101/2024.07.01.600583 (2024)
44. Ilzhöfer, D., Heinzinger, M. & Rost, B. SETH predicts nuances of residue disorder from protein embeddings. *Front. Bioinform.* **2,** 1019597 (2022).
45. Klausen, M. S. *et al.* NetSurfP□2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins* **87,** 520–527 (2019).
46. Elnaggar, A. *et al.* ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. Preprint at http://arxiv.org/abs/2007.06225 (2021)
47. Xu, M. *et al.* PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding. Preprint at https://doi.org/10.48550/arXiv.2206.02096 (2022)
48. Malladi, S., Lyu, K., Panigrahi, A. & Arora, S. On the SDEs and Scaling Rules for Adaptive Gradient Algorithms. Preprint at http://arxiv.org/abs/2205.10287 (2023)

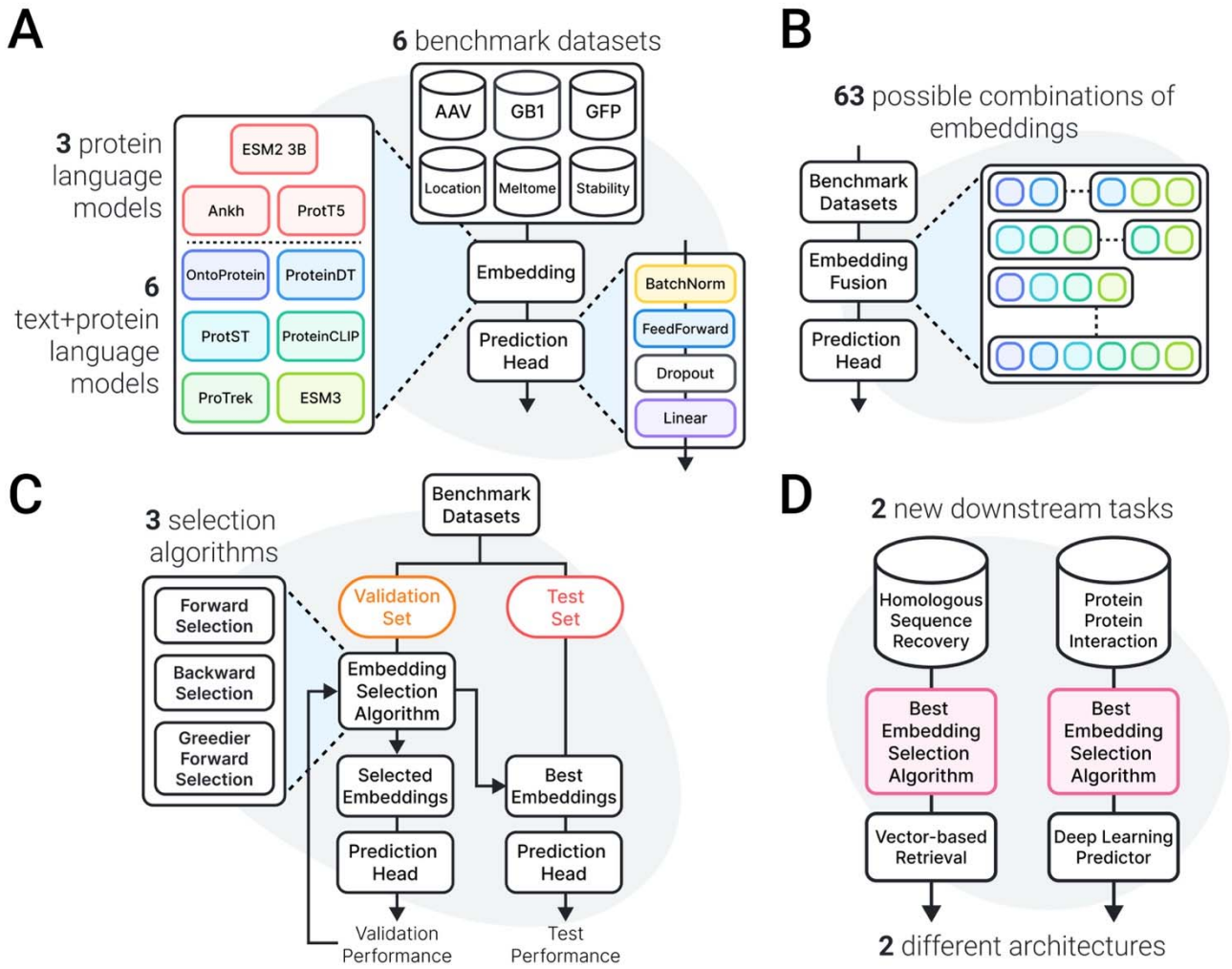


Figure 1. Study overview and benchmarking results. **A.** Six different text-integrated protein language models (OntoProtein, ProteinDT, ProtST, ProteinCLIP, ProTrek, and ESM3) are evaluated on six different datasets. Results are compared with ESM2 3B, Ankh, and ProtT5. **B.** For each dataset, all 63 possible combinations of text-integrated protein language model embeddings are benchmarked and compared with the best single tpLM embeddings. **C.** We benchmark forward selection, backward selection, and greedier forward selection in their ability to get as close to the global optimum on the same benchmark datasets **D.** We apply embedding fusion with greedier forward selection on two new tasks–homologous sequence recovery and protein-protein interaction prediction.

A

Benchmarked Models

Name	Base pLM	Parameter Count	Embedding Size
ESM2 (3B)	N/A	3B	2560
Ankh	N/A	1.9B	1536
ProtT5	N/A	3B	1024
OntoProtein	ProtBERT	420M	1024
ProteinDT	ProtBERT	420M	1024
ProtST	ESM2	650M	512
ProteinCLIP	ESM2, ProtT5	3B, 1.2B	128
ProTrek	ESM2	640M	1024
ESM3	N/A	1.4B	1546

B

Benchmark Dataset Results

Rank	AAV	GB1	GFP	Location	Meltome	Stability
#1	ESM2 (3B) 0.770±0.004	ProtT5 0.844±0.007	OntoProtein 0.666±0.002	ProTrek 0.737±0.010	ProTrek 0.689±0.005	ESM3 0.779±0.035
#2	ProtT5 0.690±0.013	Ankh 0.838±0.004	Ankh 0.647±0.002	ProtST 0.640±0.014	ProtT5 0.674±0.003	ProtT5 0.766±0.021
#3	Ankh 0.681±0.005	ProTrek 0.837±0.004	ProTrek 0.645±0.001	ESM2 (3B) 0.627±0.011	ESM2 (3B) 0.663±0.004	ESM2 (3B) 0.733±0.015
#4	OntoProtein 0.671±0.009	ESM2 (3B) 0.833±0.004	ESM3 0.640±0.002	ProteinCLIP 0.613±0.005	OntoProtein 0.648±0.008	Ankh 0.705±0.034
#5	ESM3 0.641±0.010	ESM3 0.818±0.011	ESM2 (3B) 0.613±0.002	ProtT5 0.609±0.014	Ankh 0.590±0.013	ProtST 0.699±0.036

C

Embedding Fusion Results

	AAV	GB1	GFP	Location	Meltome	Stability
% Point Difference	6.10	1.27	0.53	-0.37	3.87	2.46
Max GPU Memory	+103 Mb 2.58×	+132 Mb 3.73×	+49 Mb 1.74×	+24 Mb 1.09×	+194 Mb 3.97×	+55 Mb 1.61×
Runtime	+36.9 s 2.19×	+9.9 s 1.92×	+14.7 s 1.36×	+1.6 s 1.04×	+23.5 s 2.03×	+18.9 s 1.38×

Figure 2. Results of benchmarking text-integrated protein language models and embedding fusion. **A.** Overview of protein language models benchmarked. **B.** Benchmarking results for text-integrated protein language models and text-free protein language models. 95% CIs are shown. **C.** Percent-point difference in performance gained by using embedding fusion compared to using a single text-integrated protein language model embedding. The difference in maximum GPU memory usage and runtime (absolute difference and fold-change) between the fused embedding and single embeddings are shown.

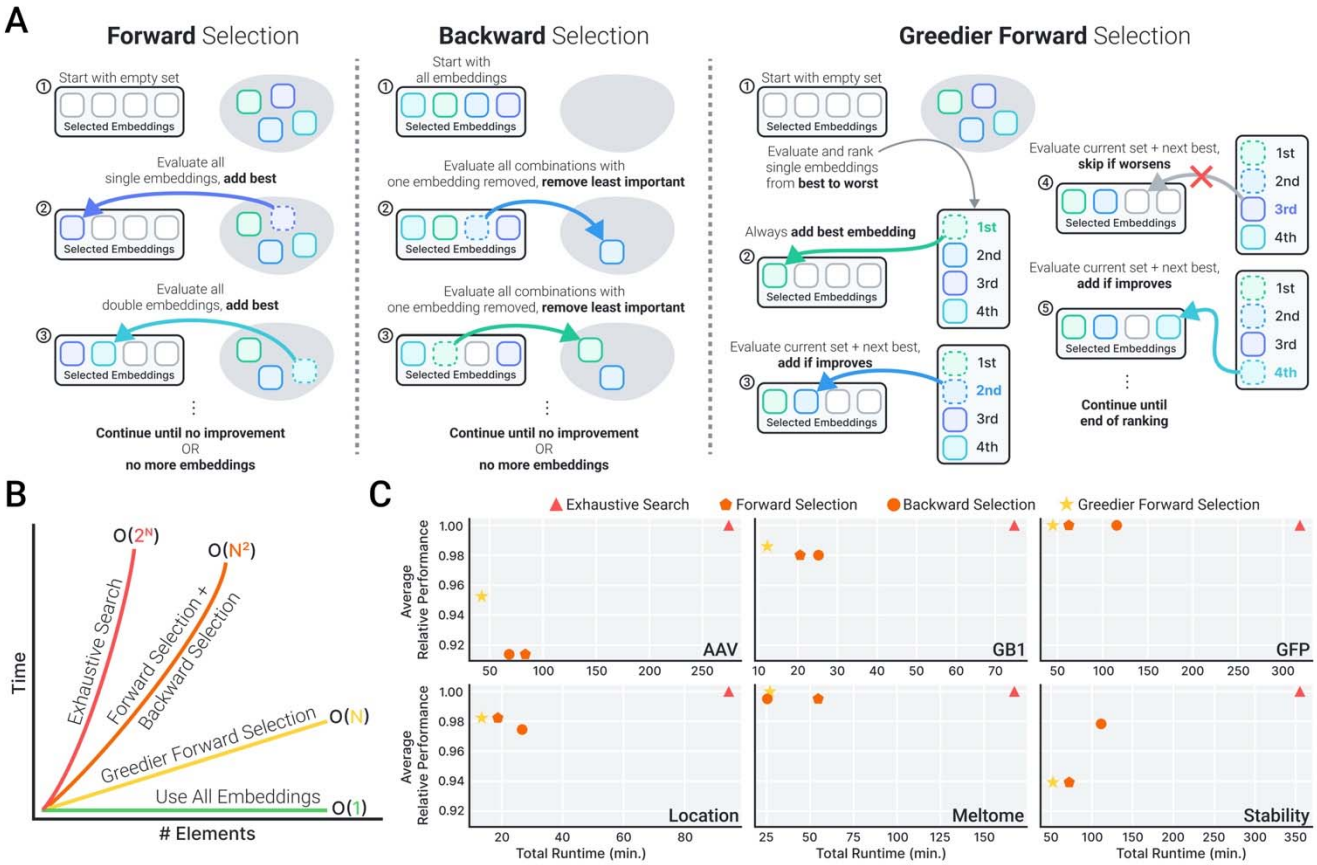
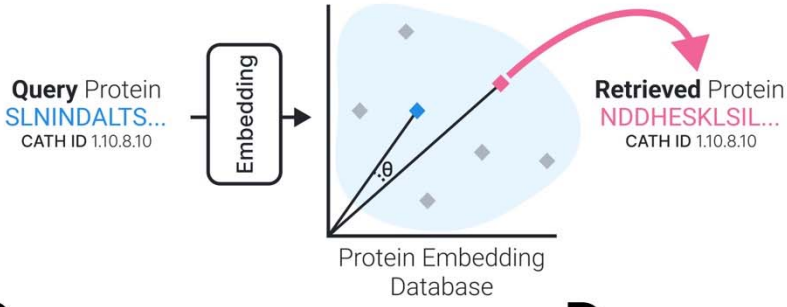


Figure 3. Benchmarking and analysis of embedding selection algorithms. **A.** Schematic overview of the embedding selection strategies evaluated: forward selection, backward selection, and greedier forward selection. **B.** Computational complexity of each selection method, visualized
alongside baseline strategies–exhaustive search and naive use of all embeddings–to highlight scalability differences. **C.** Performance comparison across six benchmark tasks, showing how each selection method approximates the optimal result from exhaustive search.

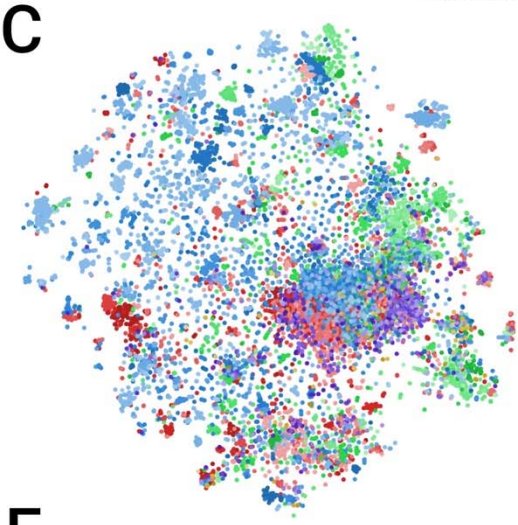
A **Vector-based** Homologous Sequence Retrieval



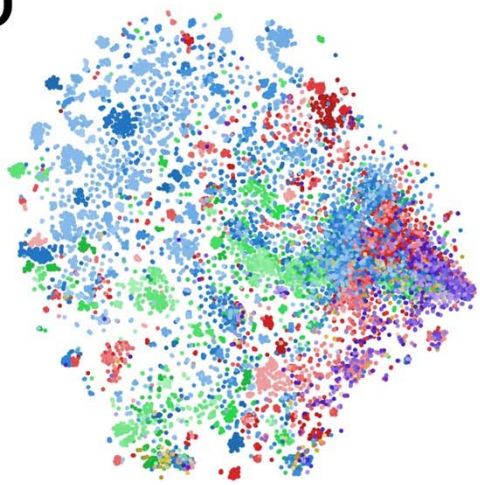
B **Retrieval** Accuracy (Top-1)

MMSeqs2	0.377
Wu	0.808
This Work	**0.831**

C



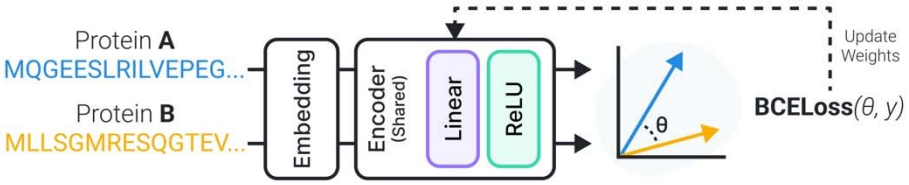
D



E **Per Class** Retrieval Accuracy

	C1 Mainly Alpha	C2 Mainly Beta	C3 Alpha-Beta	C4 Few Secondary Structures	C6 Special
Wu	1835/2676	1851/2317	5085/5825	48/98	4/10
This Work	1959/2676	1919/2317	5151/5825	50/98	5/10
Δ	**+124**	**+68**	**+66**	**+2**	**+1**

F **Protein-Protein** Interaction Predictor



G **Protein-Protein** Interaction Results

	Accuracy	AUPRC	F1	MCC	Precision	Recall	Specificity
Wu	0.651	0.697	0.680	-	-	-	-
NaderiAlizadeh	0.657	0.717	0.670	0.315	0.646	0.696	0.618
Sledzieski	0.631	0.684	0.632	0.261	0.630	0.633	0.623
Sledzieski FT	0.604	0.622	0.631	0.210	0.591	0.676	0.532
Sledzieski LoRA	0.608	0.620	0.666	0.230	0.580	**0.780**	0.436
This Work (Greedier Forward Search)	0.682	0.747	0.679	0.365	**0.687**	0.671	**0.693**
This Work (Exhaustive Seach)	**0.685**	**0.749**	**0.686**	**0.370**	0.684	0.687	0.682

H **Simulated** Experimental Validation

	Top-K	Precision	False Positives
Naderi Alizadeh	10	1.000±0.000	0.0±0.0
	100	0.968±0.016	3.2±1.6
	1000	0.909±0.002	91.0±2.4
This Work (GFS)	10	1.000±0.000	0.0±0.0
	100	**0.994±0.005**	**0.6±0.5**
	1000	**0.960±0.001**	**39.8±1.3**

Figure 4. Case study design and outcomes for homologous sequence recovery and protein-protein interaction (PPI) prediction. **A.** Overview of the homologous sequence recovery task: embeddings are used to retrieve homologous domains based on cosine similarity. **B.** Top-1 accuracy across individual and fused embeddings for the recovery task, highlighting improvements from embedding fusion. **C.** t-SNE visualization of CATH domain embeddings using the previous state-of-the-art method (ProteinCLIP). **D.** t-SNE visualization of the fused embeddings selected via greedier forward selection, showing improved clustering of CATH classes. **E.** Comparison of per-class retrieval accuracy between ProteinCLIP and the selected fusion combination, illustrating class-specific performance gains. **F.** Performance of embedding fusion on the PPI prediction task. **G.** Precision and false positives for top-k ranked predictions, simulating experimental prioritization in a resource-limited setting.