LEARNING TO READ PROTEIN LANGUAGE MODELS: INPUT-CONDITIONED FUSION OF LAYERWISE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein language models (PLMs) learn structure- and function-related representations through pretraining, and recent advances in structure-aware and multimodal modeling have further enriched their representational capabilities. However, existing research primarily focuses on improving representations through pretraining or encoder enhancement, while downstream applications typically rely on the final layer or fixed layer readout strategies, with limited systematic investigation into how to fully exploit existing multilayer information without modifying the pretrained encoder. Through layerwise evaluation, we find that the final layer is not the optimal source of representations for multiple downstream tasks, and that effective readout locations vary across models and tasks, indicating the need to distinguish pretrained models' representational capabilities from their downstream readout strategies. Based on this observation, we propose a protein-specific adaptive layer fusion method, which learns a mapping from multilayer protein representations to layer weights using supervision from the target task, producing corresponding fused representations for different inputs while keeping the pretrained encoder frozen. Our evaluation across multiple representative PLMs, three datasets and four tasks shows that reusing multilayer representations can improve over final-layer readout, and when applied to base PLMs, our method can outperform models improved through encoder enhancement. As our findings indicate, the downstream capabilities of PLMs depend not only on what representations are learned, but also on how these representations are read out for the target task, with representation learning and representation readout constituting two interrelated research dimensions.

1 INTRODUCTION

Protein language models (PLMs) learn representations related to protein structure and function through large-scale self-supervised pretraining (Rao et al., 2021; Xiao et al., 2025), providing an important foundation for protein structure prediction, remote homology detection, and function prediction (Elnaggar et al., 2022; Lin et al., 2023; Kaminski et al., 2023). To capture richer biological information, research has expanded from sequence modeling to structure-aware and multimodal learning (Hsieh et al., 2025): SaProt incorporates amino acids and discrete structural information into a unified vocabulary, S-PLM enhances structure-related representations through contrastive learning between sequences and structures, and ProTrek further aligns sequences, structures and functional text, supporting cross-modal protein retrieval and analysis (Su et al., 2024; Wang et al., 2025; Su et al., 2026). **As pretrained representations become increasingly rich, this raises a complementary question: how can the information distributed across model layers be effectively read out for downstream tasks?**

Research on existing PLM representations has followed two related directions: analyzing how information varies across layers and developing methods to reuse or combine pretrained representations. For example, Joeres et al. (2026) and Beshkov and Malthe-Sørenssen (2026) both find that the performance of PLM layers depends on the task and dataset, that the final layer does not always provide the best representation, and structural information is often stronger in layers immediately before it. Regarding representation utilization, Senoner et al. (2025) extract task-relevant signals from fixed

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.

embeddings through a supervised network; Reverse Distillation (Catrina et al., 2026) reorganizes representations from large models to improve downstream prediction. Furthermore, KinForm (Alwer & Fleming, 2026) selects intermediate layers and applies weighted pooling based on binding sites, and ESMFold (Lin et al., 2023) fuses multilayer ESM-2 representations using learnable shared weights. Together, these studies suggest that downstream performance depends on how information from different layers is selected and combined, highlighting the importance of layerwise readout for downstream tasks (Senoner et al., 2025).

However, fixed-layer selection restricts information readout to a single network depth, while shared layer weights apply the same relative weighting across layers to all inputs. Neither approach adapts the combination of layers to each protein. This raises a central question: can input-dependent layer fusion provide additional gains, and are these gains consistent across models and tasks? To address this question, we propose DynamicGate, a protein-specific adaptive layer fusion method that learns input-dependent weights over model layers. Our method shows that the downstream performance of PLMs depends not only on the representations learned by the encoder but also on how these representations are read out.

Our main contributions are as follows:

- We propose a protein-specific adaptive layer fusion method. The method adaptively organizes multilayer information for different proteins while keeping the pretrained encoder frozen, enabling task-specific representation adaptation for existing PLMs without updating the encoder.
- We analyze the role of protein-specific layer weights through mechanistic experiments. These experiments validate the effectiveness of input-dependent fusion of layerwise representations and identify its task dependence, providing evidence to clarify the conditions under which adaptive layer fusion is effective and the limits of its applicability.
- We extend multilayer representation readout to structure-aware and multimodal PLMs and find that multilayer representations remain useful even after encoder enhancement. These results suggest that encoder enhancement does not eliminate the need for effective representation readout, and that the two provide complementary routes to improving downstream performance.

2 METHODS

We propose a protein-specific adaptive layer fusion method that learns how to combine representations from multiple layers for each protein while keeping the pretrained encoder frozen. For each protein, the method extracts a representation from every layer, predicts layer weights using a shared scoring network, and combines the layer representations through a weighted sum. The gate is trained with supervision from the target task.

2.1 MULTI-LAYER PROTEIN REPRESENTATION EXTRACTION

Given a protein sequence x_i, a frozen pretrained PLM is used to extract residue representations from each Transformer layer. The output of layer l is denoted by $\mathbf{H}_i^{(l)} \in \mathbb{R}^{n_i \times d}$, where n_i is the number of residues and d is the hidden dimension. For the primary PLMs, mean pooling over valid residue positions, excluding special tokens and padding positions, yields the protein-level representation at each layer:

$$\mathbf{h}_i^{(l)} = \frac{1}{n_i} \sum_{t=1}^{n_i} \mathbf{H}_{i,t}^{(l)}, \qquad l = 1, \ldots, L. \tag{1}$$

Here, L is the number of Transformer layers in the encoder. This operation converts sequences of different lengths into fixed-dimensional protein representations while retaining the layer dimension, yielding a collection of multi-layer representations $\{\mathbf{h}_i^{(l)}\}_{l=1}^{L}$, which serves as input to the subsequent fusion network. Only amino acid sequences are provided as input throughout extraction, and the pretrained encoder parameters remain frozen.

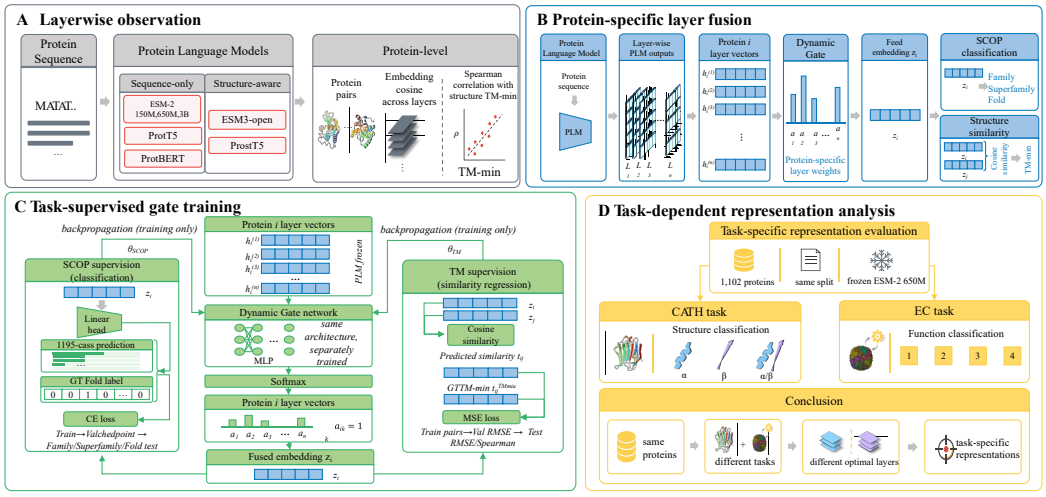


Figure 1: **Framework for multilayer representation evaluation and protein-specific adaptive fusion.**(A) Layerwise evaluation of the correlation between representation similarity and structural similarity. (B) protein-specific layer fusion and downstream prediction. (C) training the fusion network under structural similarity or classification supervision while keeping the encoder frozen. (D) comparison of the CATH and EC tasks using the same proteins, encoder, and data splits to examine the task dependence of readout strategies.

2.2 Protein-specific Adaptive Layer Fusion

Fixed-layer readout uses the same layer for all proteins within a task, whereas shared-weight fusion learns a set of input-independent layer weights through task supervision. We model layer weights as a function of the input representations, allowing different proteins within the same task to use different combinations of layers (Heddes et al., 2025).

For the representation at layer l, the shared scoring network $g_\theta : \mathbb{R}^d \to \mathbb{R}$ produces a scalar score, followed by softmax normalization across layers:

$$s_i^{(l)} = g_\theta\left(\mathbf{h}_i^{(l)}\right), \qquad \alpha_i^{(l)} = \frac{\exp\left(s_i^{(l)}\right)}{\sum_{k=1}^{L} \exp\left(s_i^{(k)}\right)}. \tag{2}$$

Normalization ensures that the layer weights are nonnegative and sum to 1. A weighted sum of the multi-layer representations is then computed to obtain a fused representation for the target task:

$$\mathbf{z}_i = \sum_{l=1}^{L} \alpha_i^{(l)} \mathbf{h}_i^{(l)}. \tag{3}$$

The scoring network parameters θ are shared across layers and proteins, whereas the layer weights $\alpha_i^{(l)}$ depend on the representation of the current protein. Protein specificity therefore resides in the fusion weights, while the network learns a shared mapping that can be applied to unseen proteins. The fused representation $\mathbf{z}_i \in \mathbb{R}^d$ retains the original hidden dimension and can be used for inter-protein similarity computation or downstream classification.

2.3 Task Supervision and Optimization

Layer weights are learned from prediction errors on the target task, without specifying in advance which layers each protein should use. Supervision is propagated through the fused representation to the scoring network, allowing the combination of layers to adapt to the target task. A separate fusion network is trained for each task, while the pretrained encoder remains frozen throughout.

For the continuous structural similarity task (Hamamsy et al., 2024; Danwada et al., 2026; Ma et al., 2026), fused representations are generated separately for the proteins in each pair (i, j), and their

cosine similarity is used as the prediction:

$$\hat{y}_{ij} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|_2 \, \|\mathbf{z}_j\|_2}. \tag{4}$$

The ground-truth structural similarity TMmin_{ij} provides supervision, and a regression loss between the predicted and target values is used to optimize θ. No additional prediction head is used for this task, supervision directly constrains the similarity between the fused representations.

For classification tasks, the fused representation is passed to a linear prediction head $f_\phi(\mathbf{z}_i) = \mathbf{W}\mathbf{z}_i + \mathbf{b}$. A classification loss is computed using the ground-truth labels, and the fusion network parameters θ and prediction head parameters ϕ are optimized jointly. The linear prediction head propagates task errors to the fusion network, allowing the combination of layers to adapt to the discriminative requirements of the target task. During inference, all trained parameters are fixed. The fusion network generates layer weights from the multi-layer representations of unseen proteins and produces predictions without requiring test labels or additional parameter updates.

3 Experimental Setup

3.1 Protein Language Models and Representation Extraction

We select ESM-2 (150M, 650M and 3B), ProtBERT, ProtT5 (Elnaggar et al., 2022), ProstT5 (Heinzinger et al., 2024) and ESM3-open (Hayes et al., 2025) for our primary evaluation, covering different model scales, architectures and pretraining approaches. We further include the structure-aware model S-PLM V1 and the multimodal model ProTrek-650M developed through enhanced pretraining of the base model ESM-2 650M, examining the applicability of the fusion method to models with enhanced representations. Pretraining information and representation extraction configurations for each model are provided in Appendix Table A.3.

All models take only amino acid sequences as input, with all pretrained parameters frozen. Layer-wise representations are converted into protein-level representations using model-specific pooling protocols, which are then used for fixed-layer evaluation and multilayer fusion.

3.2 Datasets and Tasks

Our experiments cover four tasks: structural similarity estimation, remote homology classification, structural classification, and enzyme function classification. The PISCES structural similarity data are obtained from the EBA benchmark (Pantolini et al., 2024), comprising 3,026 proteins and 20,445 protein pairs with sequence identity not exceeding 30%. TMmin is used as the structural similarity label. For SCOP Remote Homology, we use the standard benchmark splits (Hou et al., 2018). The training and validation sets contain 12,312 and 736 samples, respectively, involves classification into 1,195 Fold classes, and is evaluated on the Family, Superfamily and Fold test splits. The CATH and EC datasets are curated in this study and share 1,102 proteins with identical data splits. They correspond to three-class structural classification and four-class enzyme function classification, respectively. Dataset sizes, filtering procedures, label construction and detailed splits are provided in Appendix A.1.

3.3 Baselines and Evaluation Protocol

We consider four baselines: the final layer (Last Layer), the best single layer (Best Single Layer), uniform averaging (Uniform Mean), and fusion using layer weights shared across all proteins (Global Scalar Mix). For classification tasks, all methods use a linear prediction head. Fixed single-layer baselines and Uniform Mean train only the prediction head. Global Scalar Mix jointly trains the head and the shared layer weights, whereas our method jointly trains the head and the fusion network. For continuous structural similarity prediction, the cosine similarity between representations is used directly as the prediction, without a prediction head. Spearman's rank correlation coefficient and root mean squared error (RMSE) measure ranking consistency and numerical error between the predicted similarities and TMmin values, respectively. Classification performance is evaluated using accuracy, defined as the proportion of correctly classified samples among all test samples.

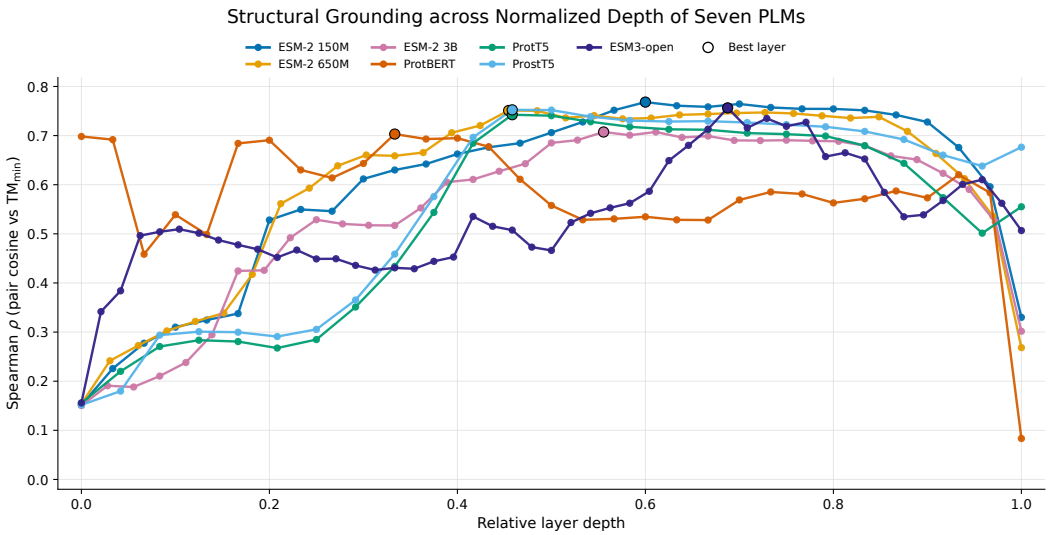


Figure 2: **Layerwise structural correlations for seven PLMs.** The x-axis shows relative layer depth, and the y-axis shows the Spearman correlation coefficient between pairwise cosine similarity of protein representations and structural similarity (TMmin). Circles mark the layer with the highest correlation for each model.

4 Experimental Results and Analysis

4.1 Effects of Interlayer Fusion on Task Representations

On the continuous structural similarity task, we find that the final layer is not the best-performing layer across the seven PLMs (Valeriani et al., 2022). Layerwise evaluation shows that all seven PLMs reach their highest correlation before the final layer (Figure 2). Additional layerwise analyses are provided in Appendix B including correlations at absolute layer depths, results stratified by sequence identity, and cross-model representation consistency. This motivates the next question: can combining multiple layers provide additional gains over the best single layer?

Table 1: Comparison of structural similarity readout performance across layer fusion strategies under TMmin supervision. Bold indicates the best result for each model. Dynamic Gate results are reported as mean ± standard deviation.

Model	Final Layer	Uniform Mean	Best Fixed Layer	Global Scalar Mix	**Dynamic Gate**
ESM-2 150M	0.3087	0.6694	0.6819	0.6604	**0.7322 ± 0.0109**
ESM-2 650M	0.1890	0.6486	0.6097	0.6587	**0.7158 ± 0.0043**
ESM-2 3B	0.2419	0.5697	0.6120	0.5844	**0.7270 ± 0.0057**
ProtBERT	0.1129	0.6446	0.5899	0.6876	**0.8139 ± 0.0054**
ProtT5	0.4455	0.6745	0.7043	0.6227	**0.7241 ± 0.0037**
ProstT5	0.6349	0.7106	0.7190	0.6571	**0.7607 ± 0.0106**
ESM3-open	0.5089	0.6880	0.7244	0.6436	**0.7992 ± 0.0023**

On PISCES, our method yields gains over single-layer selection and shared fusion in the continuous structural similarity task used to evaluate representations. These results show that task-supervised fusion of multilayer representations can improve structural similarity readout while keeping the pretrained encoder frozen. For fold classification on SCOP, we evaluate four PLMs, including sequence-only and structure-aware models. We further find that the best fixed layer for SCOP differs from that identified on PISCES. On the more challenging Fold-holdout test split, our method improves the accuracy of ESM3-open, which was pretrained using multimodal information, by 3.02 and 4.92 percentage points over the best fixed layer and Global Scalar Mix, respectively.

Table 2: Comparison of classification performance across layer fusion strategies on different SCOP generalization splits. Results are reported as mean ± standard deviation of accuracy (%). Bold indicates the best result for each model under the corresponding test setting.

Test set	Model	Final Layer	Best Fixed Layer	**Global Scalar Mix**	**Dynamic Gate**
Family holdout	ESM-2 650M	95.31 ± 0.94	97.85 ± 0.20	98.82 ± 0.16	**98.98 ± 0.14**
	ProtT5	98.45 ± 0.28	98.98 ± 0.21	99.42 ± 0.12	**99.50 ± 0.18**
	ProstT5	98.32 ± 0.55	99.00 ± 0.20	**99.19 ± 0.20**	98.98 ± 0.14
	ESM3-open	97.17 ± 0.47	98.35 ± 0.24	98.69 ± 0.35	**98.74 ± 0.28**
Superfamily holdout	ESM-2 650M	59.44 ± 0.32	76.77 ± 0.38	**79.56 ± 0.05**	79.21 ± 0.36
	ProtT5	60.69 ± 0.62	67.25 ± 1.84	72.41 ± 3.87	**73.34 ± 2.08**
	ProstT5	64.11 ± 1.59	71.05 ± 0.40	**75.09 ± 2.26**	74.03 ± 0.51
	ESM3-open	63.48 ± 0.58	78.55 ± 0.76	75.01 ± 1.55	**78.76 ± 0.97**
Fold holdout	ESM-2 650M	26.74 ± 0.91	32.68 ± 1.29	**37.28 ± 0.21**	37.05 ± 1.21
	ProtT5	26.51 ± 1.08	28.60 ± 0.98	31.15 ± 1.65	**31.52 ± 0.35**
	ProstT5	**34.26 ± 0.96**	29.85 ± 1.29	33.75 ± 0.56	33.19 ± 0.70
	ESM3-open	32.31 ± 1.51	36.07 ± 1.00	34.17 ± 0.79	**39.09 ± 0.98**

Notably, Dynamic Gate does not consistently provide additional gains in SCOP classification. Its additional gains depend on the task and model (Table 2). For example, on the Fold-holdout split, Dynamic Gate underperforms Global Scalar Mix by 0.23 percentage points for ESM-2 650M and the final-layer baseline by 1.07 percentage points for ProstT5. Overall, reusing multilayer representations can further improve task performance. Dynamic Gate shows consistent advantages across multiple PLMs on PISCES, whereas the relative gains from different fusion strategies on SCOP vary across models and test splits. These findings highlight the value of further exploiting multilayer information, while showing that the additional gains from protein-specific fusion depend on the model and task (Kabir et al., 2024). This motivates us to further examine the dependence of fusion strategies on task supervision.

4.2 Task Dependence of Fusion Strategies

As our results indicate, fusion strategies that are effective on the source task do not necessarily retain their advantages on a new task. Specifically, when Dynamic Gate trained on PISCES under TMmin supervision is directly transferred to SCOP (with the encoder and Dynamic Gate frozen, and only a linear classifier trained), its advantage on the source task does not consistently carry over (Li et al., 2024). In contrast, retraining Dynamic Gate with SCOP labels can significantly improve performance relative to frozen transfer, and the analysis of transfer from structural similarity to Fold classification is provided in Appendix C.1. This suggests that, for layer combinations suited to continuous structural similarity readout, suitability for Fold classification is not guaranteed, and to substantiate this finding, namely the importance of target-task supervision or source-task initialization, we conduct an independent retrieval experiment on the SCOP dataset, with the results supporting task-specific training of our method. Details are provided in Appendix C.2.

As the preceding comparison changes both the data and the prediction task, we further examine the effect of the supervision target using a shared set of proteins. Specifically, we keep the ESM-2 650M encoder fixed, use identical data splits for the same 1,102 proteins, and perform CATH structural classification and EC functional classification separately. For the two tasks, we observe that the best fixed layers selected are L17 and L13 respectively. Even when the input proteins and encoder remain unchanged, the effective readout location still varies with the prediction target, supporting the task dependence of representation utilization (Prabakaran & Bromberg, 2026; Joeres et al., 2026).

In this controlled setting, our method outperforms the best fixed layer and shared fusion (Table 3). These results support learning layer combinations for the target task and highlight the need to distinguish the cross-task applicability of a fusion method from the direct transferability of a trained fusion strategy. Our method thus provides a learnable approach to adapting representations to different tasks.

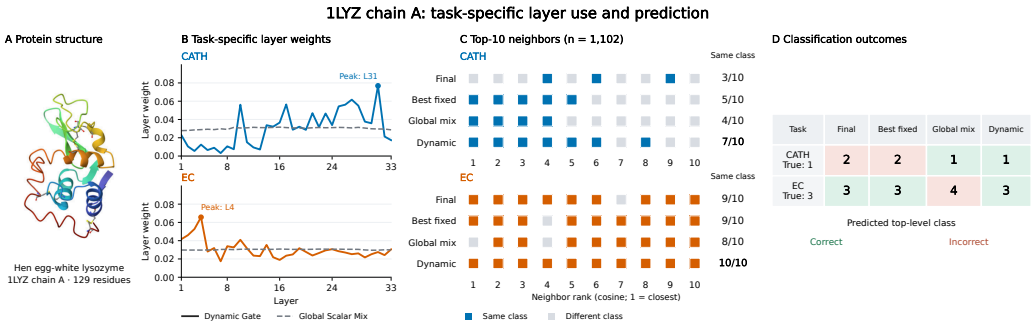


Figure 3: **Task-dependent layer fusion for hen egg-white lysozyme (PDB: 1LYZ, chain A).** (A) Protein structure. (B) Layer weights learned for CATH and EC, with solid and dashed lines denoting Dynamic Gate and Global Scalar Mix, respectively. (C) Top-10 nearest neighbors under different readout strategies, same-class neighbors are shown in color and different-class neighbors in gray. (D) Classification predictions, with green and red indicating correct and incorrect results, respectively.

Hen egg-white lysozyme (PDB: 1LYZ, chain A) provides a representative example of task-dependent layer utilization. For the same protein, Dynamic Gate assigns greater importance to deeper layers on CATH, with the highest weight at layer 31, whereas on EC the largest weight is assigned to the earlier layer 4. These different layer combinations also produce different local neighborhoods: Dynamic Gate retrieves 7/10 same-class neighbors on CATH and 10/10 on EC, and yields correct predictions for both tasks. This case illustrates that the layers most useful for representing the same protein can vary with the downstream task.

Table 3: Performance comparison of layer fusion strategies on the CATH and EC classification tasks. Results are reported for ESM-2 650M only.

Task	Final Layer	Best Fixed Layer	Global Scalar Mix	**Dynamic Gate**
CATH	0.8848	0.9515	0.9394	**0.9697**
EC	0.6242	0.6788	0.6909	**0.7273**

4.3 Mechanistic Analysis of Protein-Specific Layer Weights

To investigate where the gains from adaptive fusion come from, we conduct mechanism analyses using ESM-2 650M as a representative model. We first compare Mean-Gate, Top-4/8 Gate, and Dynamic Gate to examine the effects of shared weighting, sparse layer selection, and protein-specific weighting. Mean-Gate and Top-4/8 Gate are trained from scratch under their corresponding layer-weight constraints, and the results on PISCES and SCOP are reported in Appendix Tables D.1-D.2. The results show that Mean-Gate provides a strong baseline, while Top-8 Gate achieves a slightly higher Spearman correlation than Dynamic Gate on PISCES. These findings indicate that the joint use of multilayer information is an important source of performance gains, whereas the additional benefit of protein-specific weighting depends on the task and evaluation metric. These results also show that shared layer fusion retrained on the target task provides a strong baseline, emphasizing the importance of relearning layer combinations for each task while keeping the pretrained encoder frozen.

Furthermore, we independently retrain Dynamic Gate with three random seeds and compare between-protein weight differences, variability across runs, and weight consistency for the same protein across runs (Figure 4). As the results show, between-protein differences generally exceed seed-induced variability, while the same protein exhibits consistent weight patterns across training runs, indicating that the model learns reproducible protein-specific layer weights. Subsequently, we randomly permute the layer-weight vectors across proteins without retraining the model. Permutation has a markedly greater effect on PISCES than on SCOP, further indicating that the effect of protein-specific weighting depends on the task.

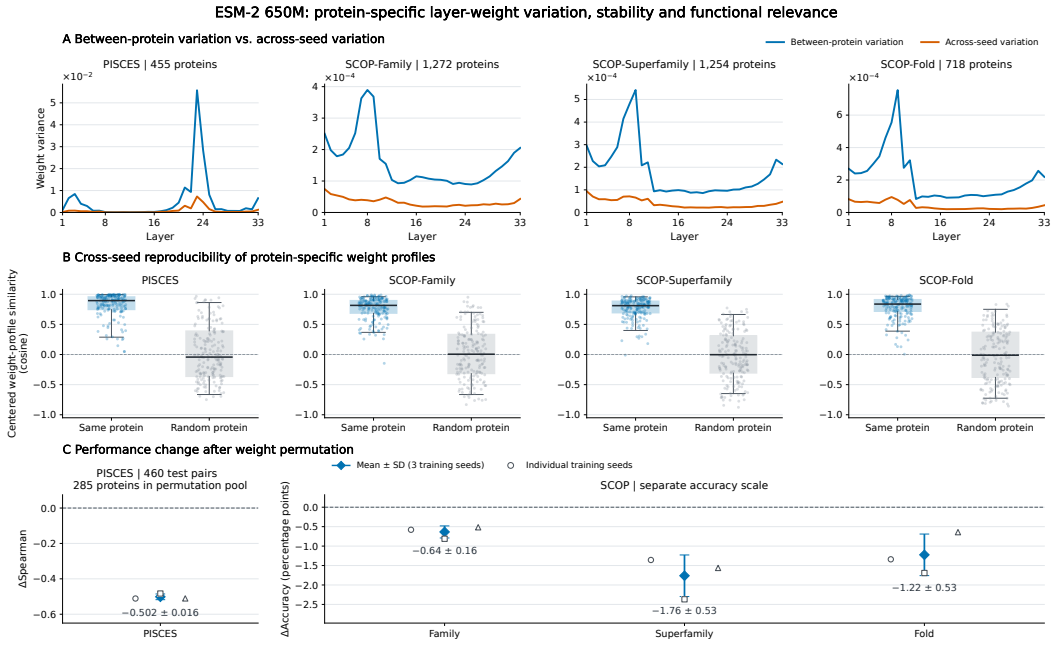


Figure 4: **Variability, stability, and functional role of protein-specific layer weights in ESM-2 650M.** (A) Comparison of weight differences between proteins with weight variability across independent training runs using different random seeds. (B) comparison of weight-pattern similarity across independent training runs for the same protein versus randomly paired proteins, assessing the reproducibility of protein-specific weights. (C) comparison of task performance before and after randomly permuting layer weights across proteins without retraining the model.

In addition, using the same ESM-2 650M encoder, we conduct control experiments including within-class and between-class weight swaps (Appendix D.3–D.4), and a parameter-matched MLP. These experiments examine the role of weight matching while controlling for additional parameters. For the SCOP Fold classification task, we further introduce two MLP controls: Final Layer + MLP and All Layers Concatenation + MLP, with results reported in Appendix Table D.3. Overall, dynamically adjusting layer weights provides greater performance improvements on downstream tasks than applying a standard nonlinear transformation to fixed representations.

5 Encoder Enhancement and Representation Readout

We further apply protein-specific adaptive layer fusion to the structure-aware model S-PLM and the multimodal model ProTrek-650M (implementation details are provided in Appendix Table E.1), both based on ESM-2 650M, to examine whether, after their representations have been enhanced through pretraining, such enhancement can translate directly into better downstream performance or whether performance remains influenced by the representation readout strategy.

Overall, structure-aware and multimodal pretraining do not eliminate the benefits of multilayer representation readout. By selecting or combining representations from different layers, both S-PLM and ProTrek-650M can outperform final-layer readout on multiple tasks. This indicates that task-relevant information remains distributed across different layers even after enhanced pretraining, and that the final layer does not fully capture the information available within the model. However, on some classification tasks, Best Fixed Layer or Global Scalar Mix can match or outperform Dynamic Gate. These results distinguish the benefit of using multilayer representations from the benefit of protein-specific dynamic fusion, with the latter depending on both the model and the task.

This raises a further question of particular interest: how does representation enhancement at the encoder level relate to representation optimization at the readout stage? S-PLM is an enhanced version

Table 4: Comparison of representation readout performance for S-PLM and ProTrek-650M on representative tasks. Complete PISCES RMSE and SCOP Family/Fold results are provided in Appendix Tables E.2-E.3. Boldface indicates the best result for each model on the corresponding task.

Model	Readout	PISCES Spearman ↑	SCOP Superfamily ↑	CATH ↑	EC ↑
S-PLM	Final Layer	0.7397	63.37 ± 0.32	89.70 ± 1.60	67.47 ± 0.93
	Best Fixed	0.6222	72.59 ± 1.08	92.93 ± 2.45	**73.33 ± 1.60**
	Global Scalar Mix	0.6601	70.39 ± 3.02	93.13 ± 0.93	69.70 ± 1.60
	Dynamic Gate	**0.7453 ± 0.0038**	**73.42 ± 1.43**	**94.95 ± 0.93**	68.28 ± 0.93
ProTrek-650M	Final Layer	0.4175	48.27 ± 0.12	80.00 ± 1.21	65.86 ± 0.93
	Best Fixed	0.5504	**64.97 ± 0.09**	83.64 ± 2.64	**73.54 ± 1.26**
	Global Scalar Mix	0.5509	64.11 ± 0.70	86.67 ± 0.61	71.72 ± 2.80
	Dynamic Gate	**0.7212 ± 0.0184**	63.85 ± 0.12	**86.87 ± 2.29**	70.10 ± 0.93

of ESM-2 650M, providing a more direct basis for comparison. We therefore further compare the performance of ESM-2 650M and S-PLM on the same SCOP task under different readout strategies.

Table 5: Encoder enhancement and representation readout: performance comparison of ESM-2 650M and S-PLM across the three SCOP generalization splits.

Test set	ESM-2 Final Layer	S-PLM Final Layer	ESM-2 Best Fixed	**ESM-2 Dynamic Gate**	S-PLM Dynamic Gate
Family holdout	95.31 ± 0.94	98.40 ± 0.09	97.85 ± 0.20	**98.98 ± 0.14**	98.35 ± 0.31
Superfamily holdout	59.44 ± 0.32	63.37 ± 0.32	76.77 ± 0.38	**79.21 ± 0.36**	73.42 ± 1.43
Fold holdout	26.74 ± 0.91	35.65 ± 0.50	32.68 ± 1.29	**37.05 ± 1.21**	33.43 ± 2.19

This comparison reveals an important pattern (Table 5): S-PLM outperforms its baseline, ESM-2 650M, under final-layer readout, whereas ESM-2 650M performs better when multilayer readout is applied. This is consistent with the pattern observed for ESM3-open. One possible explanation is that encoder enhancement redistributes task-relevant information across layers and alters the complementarity between them. This observation indicates that better final-layer performance after encoder enhancement does not necessarily imply greater utility across the full layer hierarchy. Therefore, encoder enhancement and multilayer representation readout provide distinct ways to improve downstream performance.

6 Discussion and Conclusion

This study investigates multilayer representation readout in protein language models. Our experiments show that the final layer is not the best readout choice for most models and that reusing multilayer representations with a frozen encoder can improve downstream performance. However, the best readout strategy varies across models and tasks, and the benefits of protein-specific adaptive layer fusion show the same dependence. Further analyses of S-PLM and ProTrek-650M show that multilayer representation readout remains useful after encoder enhancement, and that better final-layer performance does not necessarily imply greater utility across the full layer hierarchy. These findings indicate that representation learning and representation readout are related but distinct. The current experiments cover a limited number of models and task types. Future work will extend the evaluation to a broader range of sequence-only, structure-aware, and multimodal PLMs and prediction tasks. We will further investigate how encoder enhancement changes the distribution of information across layers and how these changes relate to readout strategies, with the goal of better characterizing the applicability of adaptive multilayer representation readout.

AI USE STATEMENT

Generative AI tools were used solely to assist with editing and polishing the manuscript, including improving grammar, clarity, concision, organization, and readability. They were not used to formulate the research questions or hypotheses, design the methodology or experiments, generate or analyze data, develop mathematical claims or proofs, or interpret the results. All scientific content, experimental procedures, analyses, results, and conclusions were developed and verified by the authors. The authors reviewed and approved all AI-assisted edits and take full responsibility for the final manuscript.

ETHICS STATEMENT

This study evaluates protein language model representations using protein sequences, structures, and annotations from public resources. The work is entirely computational and does not involve human participants, animal experiments, or the collection of personally identifiable information.

REPRODUCIBILITY STATEMENT

We provide an anonymized supplementary code package containing the implementations of the readout methods and ablation experiments reported in this paper, together with task-specific configurations, fixed data splits, evaluation scripts, and selected experimental results. Experimental settings, dataset construction and splits, optimization details, and backbone-specific protocols are documented in the appendix and supplementary materials. Small data files required for reproduction are included, while raw protein structures, pretrained model weights, and large cached embeddings are not redistributed; instructions for obtaining and preparing the required public datasets and pretrained models are provided in the supplementary materials. Random seeds and evaluation procedures are specified in the configuration files to facilitate reproduction of the reported experiments.

REFERENCES

Saleh Alwer and Ronan M. T. Fleming. KinForm: kinetics-informed feature optimised representation models for enzyme k_{cat} and K_{M} prediction. *npj Systems Biology and Applications*, 12(1): 71, 2026. doi: 10.1038/s41540-026-00692-5.

Kosio Beshkov and Anders Malthe-Sørenssen. Towards understanding the shape of representations in protein language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Darius Catrina, Christian Bepler, Samuel Sledzieski, and Rohit Singh. Reverse distillation: Consistently scaling protein language model representations. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shivaram Danwada, Priscilla Udomprasert, Nilanjana Raychawdhary, Cheryl D Seals, Lei Wu, and Sutanu Bhattacharya. Evaluating protein language model embeddings for structural similarity in the protein-sequence twilight zone. *bioRxiv*, 2026. doi: 10.64898/2026.07.31.742099.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/tpami.2021.3095381.

Tymor Hamamsy, James T. Morton, Robert Blackwell, Daniel Berenberg, Nicholas Carriero, Vladimir Gligorijevic, Charlie E. M. Strauss, Julia Koehler Leman, Kyunghyun Cho, and Richard Bonneau. Protein remote homology detection and structural alignment using deep learning. *Nature Biotechnology*, 42(6):975–985, 2024. doi: 10.1038/s41587-023-01917-2.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra,

Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018.

Mike Heddes, Adel Javanmard, Kyriakos Axiotis, Gang Fu, Mohammadhossein Bateni, and Vahab Mirrokni. DeepCrossAttention: Supercharging transformer residual connections. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22881–22903. PMLR, 2025. URL https://proceedings.mlr.press/v267/heddes25a.html.

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

Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018. doi: 10.1093/bioinformatics/btx780.

Cheng-Yen Hsieh, Xinyou Wang, Daiheng Zhang, Dongyu Xue, Fei Ye, Shujian Huang, Zaixiang Zheng, and Quanquan Gu. Elucidating the design space of multimodal protein language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24156–24175. PMLR, 2025. URL https://proceedings.mlr.press/v267/hsieh25a.html.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Roman Joeres, Ilya Senatorov, Anastasia Kolchina, Dietrich Klakow, and Olga V. Kalinina. Task- and dataset-specific information in protein language models. *arXiv preprint arXiv:2608.12090v2*, 2026.

Anowarul Kabir, Asher Moldwin, Yana Bromberg, and Amarda Shehu. In the twilight zone of protein sequence homology: do protein language models learn protein structure? *Bioinformatics Advances*, 4(1):vbae119, 2024. doi: 10.1093/bioadv/vbae119.

Kamil Kaminski, Jan Ludwiczak, Kamil Pawlicki, Vikram Alva, and Stanislaw Dunin-Horkawicz. pLM-BLAST: distant homology detection based on direct comparison of sequence representations from protein language models. *Bioinformatics*, 39(10):btad579, 2023. doi: 10.1093/bioinformatics/btad579.

Francesca-Zhoufan Li, Ava P Amini, Yisong Yue, Kevin K Yang, and Alex Xijie Lu. Feature reuse and scaling: Understanding transfer learning with protein language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27351–27375. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24a.html.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Wei Liu, Ziye Wang, Ronghui You, Chenghan Xie, Hong Wei, Yi Xiong, Jianyi Yang, and Shanfeng Zhu. PLMSearch: Protein language model powers accurate and fast sequence search for remote homology. *Nature Communications*, 15(1):2775, 2024. doi: 10.1038/s41467-024-46808-5.

Zinnia Ma, Javier Espinoza Herrera, Elsy Buitrago-Delgado, Neville P. Bethel, and Adrian Jinich. Inferring local protein structural similarity from sequence alone. *Journal of Chemical Information and Modeling*, 66(10):5699–5707, 2026. doi: 10.1021/acs.jcim.6c00114.

Lorenzo Pantolini, Gabriel Studer, Joana Pereira, Janani Durairaj, Gerardo Tauriello, and Torsten Schwede. Embedding-based alignment: combining protein language models with dynamic programming alignment to detect structural similarities in the twilight-zone. *Bioinformatics*, 40(1): btad786, 2024. doi: 10.1093/bioinformatics/btad786.

R. Prabakaran and Yana Bromberg. Quantifying uncertainty in protein representations across models and tasks. *Nature Methods*, 23(4):796–804, 2026. doi: 10.1038/s41592-026-03028-7.

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations*, 2021.

Tobias Senoner, Ivan Koludarov, Joshua Günther, Amarda Shehu, Burkhard Rost, and Yana Bromberg. Which pLM to choose? *bioRxiv*, 2025. doi: 10.1101/2025.10.30.685515.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2024.

Jin Su, Yan He, Shiyang You, Shiyu Jiang, Xibin Zhou, Xuting Zhang, Yuxuan Wang, Xining Su, Igor Tolstoy, Xing Chang, Hongyuan Lu, and Fajie Yuan. A trimodal protein language model enables advanced protein searches. *Nature Biotechnology*, 44(8):1310–1316, 2026. doi: 10.1038/s41587-025-02836-0.

Sanjana Tule, Gabriel Foley, and Mikael Bodén. Do protein language models learn phylogeny? *Briefings in Bioinformatics*, 26(1):bbaf047, 2025. doi: 10.1093/bib/bbaf047.

Lucrezia Valeriani, Francesca Cuturello, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of protein language models. *bioRxiv*, 2022. doi: 10.1101/2022.10.24.513504.

Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Rajani. BERTology meets biology: Interpreting attention in protein language models. In *International Conference on Learning Representations*, 2021.

Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, and Dong Xu. S-PLM: Structure-aware protein language model via contrastive learning between sequence and structure. *Advanced Science*, 12(5):2404212, 2025. doi: 10.1002/advs.202404212.

Yijia Xiao, Wanjia Zhao, Junkai Zhang, Yiqiao Jin, Han Zhang, Zhicheng Ren, Renliang Sun, Haixin Wang, Guancheng Wan, Pan Lu, Xiao Luo, Yu Zhang, James Zou, Yizhou Sun, and Wei Wang. Protein large language models: A comprehensive survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23080–23103. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1255.

APPENDIX: EXPERIMENTAL SETUP AND SUPPLEMENTARY RESULTS

This appendix provides additional details on dataset construction, model configurations, training, and model selection, and reports results from layerwise analyses, transfer experiments, post-training ablations, controls trained from scratch, and interventions on layer weights. Detailed protocols and full comparison results for S-PLM and ProTrek-650M are provided in Appendix E.

A EXPERIMENTAL SETUP AND IMPLEMENTATION DETAILS

A.1 DATA SOURCES AND PREPROCESSING

Table A.1 lists the datasets used in this study and their sizes.

Table A.1: Datasets used in this study and their sizes.

Dataset	Split / Statistics	Size
PISCES	Proteins	3,026
	Protein pairs	20,445
SCOP Remote Homology	Train	12,312
	Validation	736
	Family	1,272
	Superfamily	1,254
	Fold	718
CATH / EC	Shared proteins	1,102

A.1.1 PISCES STRUCTURAL SIMILARITY DATASET

We construct a structural similarity benchmark from the retained PISCES protein sequences, protein-pair records, and their corresponding TM-scores. Protein pairs are first canonicalized as unique unordered pairs. We check the consistency of the percentage sequence identity (PIDE) and TM-score records and use the smaller of the two corresponding TM-scores as the supervision target:

$$\mathrm{TMmin}(i,j) = \min\{\mathrm{TM}_1(i,j), \mathrm{TM}_2(i,j)\}$$

The initial dataset contains 3,028 proteins and 20,466 protein pairs, with PIDE no greater than 30% for each retained pair. To meet the length requirements for representation extraction, we remove two proteins exceeding 1,022 residues and the 21 pairs involving them. The final benchmark contains 3,026 proteins and 20,445 protein pairs.

The full layerwise analysis uses all 20,445 protein pairs. For fusion network training and evaluation, we split the dataset at the protein level using random seed 20260902 (Table A.2). Only pairs with both proteins in the same subset are retained, the remaining 9,189 pairs spanning different subsets are excluded.

Table A.2: Numbers of proteins and protein pairs in each subset for the PISCES fusion experiments.

Data Split	**Number of Proteins**	**Number of Retained Protein Pairs**
Training Set	2,118	10,384
Validation Set	453	412
Test Set	455	460

A.1.2 SCOP REMOTE HOMOLOGY CLASSIFICATION DATASET

We use the SCOP Remote Homology dataset provided by TAPE, retaining the official splits and label encoding. Source: the TAPE code repository. This is a single fold classification task with 1,195 classes. The training and validation sets contain 12,312 and 736 proteins, respectively. The Family-, Superfamily-, and Fold-holdout test sets contain 1,272, 1,254, and 718 proteins, respectively. These are three holdout sets defined by homology relationships within the same classification task, rather than three distinct prediction targets.

A.1.3 CATH/EC DATASET WITH SHARED PROTEINS

To compare the representation requirements of structural and functional tasks for the same proteins, we construct a shared protein dataset using 15,051 protein sequences from the trRosetta series and their corresponding PDB structures retained in our project. The corresponding public resource is Single(MSA)15051 from trRosettaX-Single. Source: the official trRosettaX-Single data page.

Stage 1: Quality Control of the Source Dataset

We first check consistency between sequences and structural coordinates. Records are excluded if the corresponding structure is missing, the original chain cannot be identified, the ratio of fragment length to original chain length is below 90%, or the sequence exceeds 1,022 residues. This filtering yields 11,048 candidate proteins. The 90% criterion refers to the length ratio rather than separately estimated sequence alignment coverage.

Stage 2: Removal of Records Overlapping with or Homologous to PISCES Proteins

Candidate records are removed if they meet any of the following criteria: sharing a PDB chain identifier with a PISCES protein, having an identical sequence to a PISCES protein, or yielding a homology search hit with PIDE >30% and coverage $\geq$80% for at least one sequence (query or target). In total, 3,096 records meet at least one criterion, leaving 7,952 candidate proteins after removal.

Stage 3: Linking Structural and Functional Labels

Using PDB IDs and chain IDs, we link SIFTS chain-level CATH mappings and EC annotations from the August 30, 2026 snapshot. Structural classes are determined from the CATH v4.4.0 domain list. SIFTS provides separate mapping files linking PDB chains to CATH domains and enzyme EC numbers, respectively.

We retain proteins mapped to exactly one CATH domain and assigned to a single first-level EC class. We then restrict the dataset to CATH classes 1-3 and first-level EC classes 1-4 and remove sequences longer than 512 residues.

The final prediction tasks are:

CATH:Three-class classification of top-level structural classes, rather than topology or fold classification.

EC:Four-class classification of first-level functional classes, rather than prediction of full EC numbers.

A single first-level EC class does not require a single full EC number: a protein may have multiple full EC numbers, provided that they all belong to the same first-level class. This applies to 72 proteins in the final dataset.

Stage 4: Shared Data Splits for Both Tasks

Using split seed 42, we stratify proteins by their joint CATH $\times$ EC labels and adjust sample allocation within each stratum so that the length-bin distributions in each subset approximate the specified proportions. The final training, validation, and test sets contain 772, 165, and 165 proteins, respectively.

Both tasks use exactly the same proteins, sequences, pre-extracted features, and split assignments, but their fusion modules and classification heads are trained separately. The retained sequences range from 42 to 511 residues, with no duplicate identifiers or identical sequences.

A.2 Pretrained Models and Representation Extraction

Table A.3: Pretrained models and encoder configurations used in this study.

Model	Pretraining information	Pretrained version	Components used	Layers (L)	Hidden dimension (d)
ESM-2 150M	Sequence	`facebook/esm2_t30_150M_UR50D`	Transformer encoder	30	640
ESM-2 650M	Sequence	`facebook/esm2_t33_650M_UR50D`	Transformer encoder	33	1,280
ESM-2 3B	Sequence	`facebook/esm2_t36_3B_UR50D`	Transformer encoder	36	2,560
ProtBERT	Sequence	`Rostlab/prot_bert`	BERT encoder	30	1,024
ProtT5	Sequence	`Rostlab/prot_t5_xl_half_uniref50-enc`	T5 encoder	24	1,024
ProstT5	Sequence, 3Di structure tokens	`Rostlab/ProstT5_fp16`	T5 encoder	24	1,024
ESM3-open	Sequence, structure, function	`EvolutionaryScale/esm3-sm-open-v1`	Transformer backbone	48	1,536
S-PLM V1	Sequence, structure	the sequence branch of the official V1 checkpoint	ESM-2 encoder with structure-aware adapters	33	1,280
ProTrek-650M	Sequence, structure, text	`westlake-repl/ProTrek_650M`	Sequence encoder and native projection layer	33	1,280

Layer counts exclude the input embedding layer L0. For T5-based models, only the encoder is used, without the decoder. All pretrained backbones are frozen, as are the pretrained adapters in S-PLM and the native projection layer in ProTrek. ProTrek's native projection maps 1,280-dimensional representations to 1,024 dimensions.

A.3 Best Fixed Layer Selection

In the SCOP experiments with four models, we train a separate linear classifier for each layer. We select the fixed layer based on mean validation accuracy across three random seeds and use it unchanged across all three test sets. For CATH and EC, fixed layers are selected separately based on each task's validation accuracy with seed 42, ties are resolved in favor of the shallower layer.

B Supplementary Analyses of Representations

B.1 Complete Layerwise Structural Correlations

To examine how structure-related information varies with model depth, we compute Spearman's correlation between protein embedding cosine similarity and TMmin at each layer, using the same set of 20,445 protein pairs from 3,026 proteins. Layer positions are expressed using the original layer indices. Figure B.1 shows the complete curves plotted against these indices.

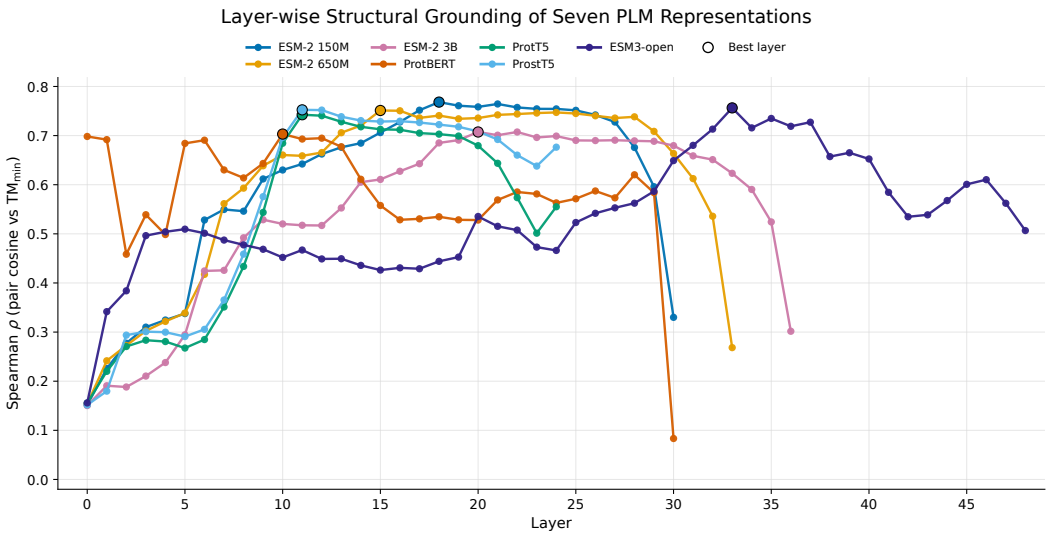


Figure B.1: **Layerwise structural correlations for seven protein language models.** At each layer, Spearman's correlation between protein-pair cosine similarity and TMmin is computed over 20,445 protein pairs from 3,026 proteins. The horizontal axis shows the original layer index, with layer 0 corresponding to the input embedding layer. Large circles mark the layer with the highest structural correlation for each model on this dataset.

B.2 Layerwise Results Stratified by Sequence Identity

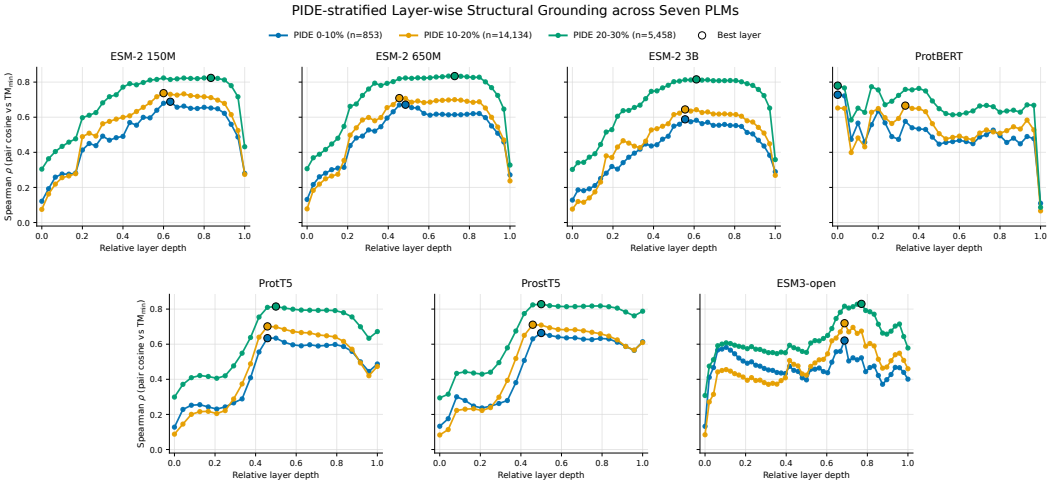


Figure B.2: **Structural correlations stratified by sequence identity**. Layerwise correlations within the PIDE intervals [0, 10)%, [10, 20)%, and [20, 30]%, containing 853, 14,134, and 5,458 protein pairs, respectively. Peak locations and correlation strengths vary across intervals. Unequal sample sizes and differing TMmin distributions limit causal interpretation of these differences.

We divide protein pairs into three PIDE intervals, [0, 10)%, [10, 20)%, and [20, 30]%, containing 853, 14,134, and 5,458 pairs, respectively, and recompute layerwise structural correlations within each interval (Figure B.2).

For all models, peak correlations in the 20–30% interval exceed those in the two lower-identity intervals, but peak layer locations do not shift in a consistent direction. For example, ESM-2 650M peaks at L16, L15, and L24 across the three intervals in ascending order. The descriptive layerwise

analysis includes the input embedding layer L0, ProtBERT peaks at L0 in both the lowest and highest PIDE intervals.

These observations indicate that both correlation strength and peak location depend on sample composition. However, sample sizes and TMmin distributions differ across intervals, so these differences should not be interpreted as causal effects of sequence identity.

B.3 Cross-Model Representation Consistency

We compare representations from the final layer and the layer with the highest structural correlation in terms of global relationships and local neighborhoods. Global consistency is measured by Spearman's correlation between the similarity rankings assigned by different models to the same 20,445 protein pairs. Local consistency is measured by Top-20 nearest-neighbor overlap: for each protein, each model selects its 20 nearest neighbors from the remaining 3,025 proteins. We divide the intersection size of the two models' neighbor sets by 20 and average over all proteins. Results are shown in Figure B.4.

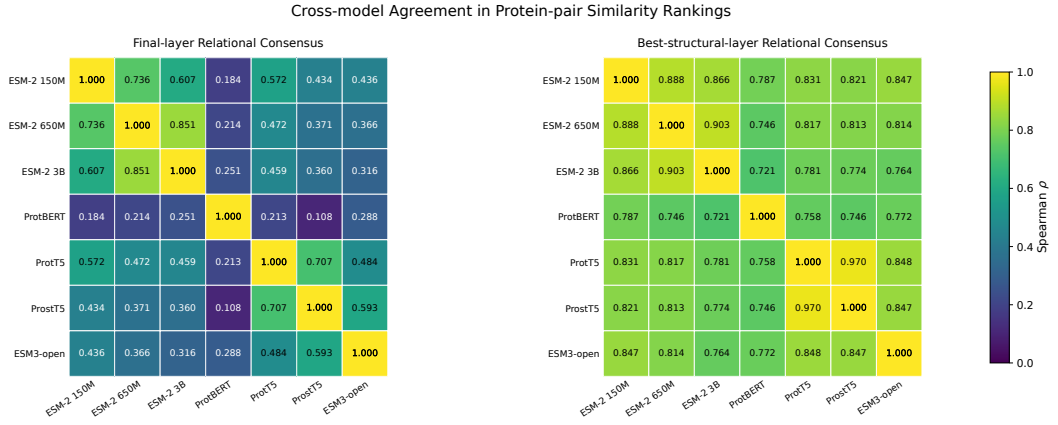


Figure B.3: **Cross-model consistency of global protein-pair similarity rankings.** Cosine similarity vectors from the final layers and the layers with the highest structural correlation are compared using the same 20,445 matched protein pairs. Reported medians exclude the matrix diagonals and are computed across 21 distinct model pairs. Best-layer selection uses a common structural target and forms part of a descriptive analysis (Huh et al., 2024).

As shown in Figures B.3 and B.4, using each model's best structural layer improves both cross-model similarity-ranking consistency and local consistency for all 21 model pairs. This indicates that cross-model agreement on protein relationships is strongly layer-dependent: the selected structural layers yield more consistent pairwise similarity rankings than the final layers. These findings show that differences observed at the final layer do not fully reflect the structure-related relationships captured elsewhere in the network, motivating the study of layer selection and fusion rather than default reliance on final-layer representations. This analysis provides descriptive evidence of cross-model consistency, rather than direct evidence that dynamic fusion outperforms fixed aggregation.

C Cross-Task Transfer and Post-Training Ablations

C.1 Frozen Transfer from Structural Similarity to SCOP

We first test whether layer preferences learned under TMmin supervision on PISCES can be applied unchanged to SCOP fold classification. For ESM-2 650M and ProtT5, we freeze the PLM and the fusion module trained on the source task, training only a 1,195-class linear classifier on SCOP. The fixed layers (L15 and L11) are selected based on the original PISCES cosine similarity–TMmin analysis rather than SCOP validation accuracy. Thus, this experiment evaluates the transfer of source-task representations rather than learning fusion on the target task.

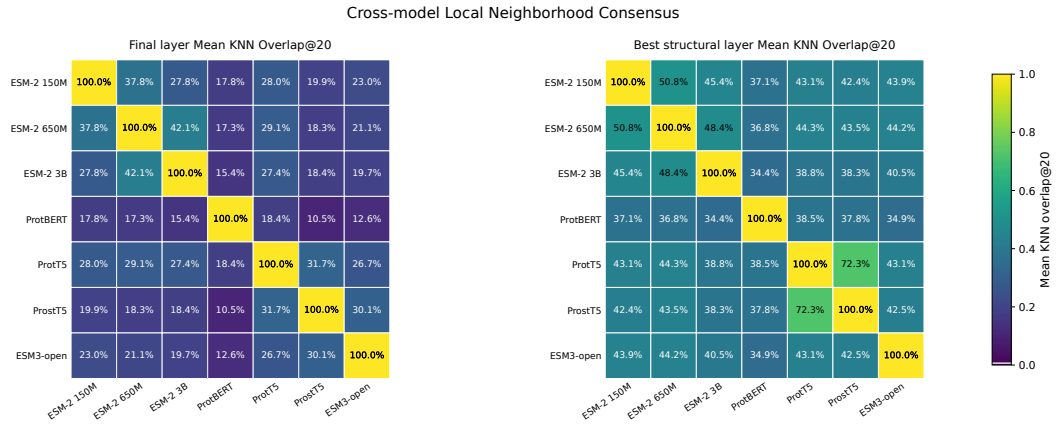


Figure B.4: **Cross-model consistency of local protein neighborhoods.** For each protein, the 20 nearest neighbors are selected from all 3,025 remaining proteins. The intersection size of two models' neighbor sets is divided by 20 and averaged over 3,026 proteins. Neighbor retrieval is not restricted to the 20,445 preselected protein pairs.

Table C.1: Frozen PISCES representations with a linear classifier trained on SCOP. Accuracy (%).

Backbone Model	Frozen Representation	Family Holdout	Superfamily Holdout	Fold Holdout
ESM-2 650M	Final Layer	95.31 ± 0.94	59.44 ± 0.32	26.74 ± 0.91
	PISCES Fixed Layer (L15)	**98.11 ± 0.28**	73.15 ± 0.61	31.06 ± 0.91
	Frozen Global Scalar Mix	97.72 ± 0.34	**74.75 ± 0.49**	**31.99 ± 1.01**
	Frozen Dynamic Gate	98.09 ± 0.16	73.52 ± 0.69	31.38 ± 0.58
ProtT5	Final Layer	98.45 ± 0.28	60.69 ± 0.62	26.51 ± 1.08
	PISCES Fixed Layer (L11)	98.69 ± 0.12	**67.36 ± 1.48**	**28.09 ± 2.11**
	Frozen Global Scalar Mix	**98.93 ± 0.30**	64.83 ± 1.59	27.62 ± 1.25
	Frozen Dynamic Gate	98.53 ± 0.28	61.51 ± 1.89	27.25 ± 0.21

The frozen Dynamic Gate outperforms Final Layer in all six combinations of backbone model and test set, but does not yield the strongest transfer representation in any combination. This indicates that the advantage of input-dependent weighting on the source task does not automatically persist under different objectives.

C.2 Target Task Adaptation: A Controlled Retrieval Study

To examine the importance of target-task supervision or source-task initialization, we conduct a separate retrieval experiment with ESM-2 650M, including three gating conditions: frozen gate parameters trained under TMmin supervision on PISCES, a gate trained from scratch on SCOP, and a gate initialized from the PISCES gate and fine-tuned on SCOP. The PLM remains frozen throughout, and no classifier is trained. In the two adaptation conditions, only the gate parameters are optimized. The two adaptation conditions use the same supervised contrastive objective, optimizer, batch sampling strategy and checkpoint selection rule, differing only in initialization.

Across all three test sets, for both gates trained on the target task, mean Recall@1 exceeds that of the frozen source-task gate. Training from scratch also yields higher mean Recall@1 than source-task initialization across all three test sets, although source-task initialization achieves the highest

Table C.2: Evaluation of gate adaptation for ESM-2 650M through fold retrieval. Query hit rate(%).

Test Split	Gate Condition	Recall@1	Recall@10	Recall@30
Family Holdout	Frozen TMmin Gate	96.59 ± 0.51	98.72 ± 0.09	99.24 ± 0.09
	SCOP Training from Scratch	**98.79 ± 0.12**	**99.32 ± 0.05**	99.61 ± 0.05
	TMmin Initialization + SCOP Training	98.58 ± 0.21	99.27 ± 0.05	**99.50 ± 0.00**
Superfamily Holdout	Frozen TMmin Gate	69.03 ± 0.60	80.28 ± 0.77	84.08 ± 0.61
	SCOP Training from Scratch	**78.50 ± 0.53**	**85.41 ± 0.08**	**88.68 ± 0.14**
	TMmin Initialization + SCOP Training	78.31 ± 0.14	85.35 ± 0.20	88.44 ± 0.21
Fold Holdout	Frozen TMmin Gate	27.11 ± 0.84	42.39 ± 0.66	51.21 ± 0.71
	SCOP Training from Scratch	**32.54 ± 0.64**	**46.84 ± 0.16**	**54.69 ± 0.64**
	TMmin Initialization + SCOP Training	31.75 ± 0.48	44.48 ± 0.16	52.37 ± 1.00

Family Recall@30. **These observations support task-specific training, but do not indicate a benefit from using TMmin pretraining for initialization.**

C.3 Training Dynamics for SCOP Classification

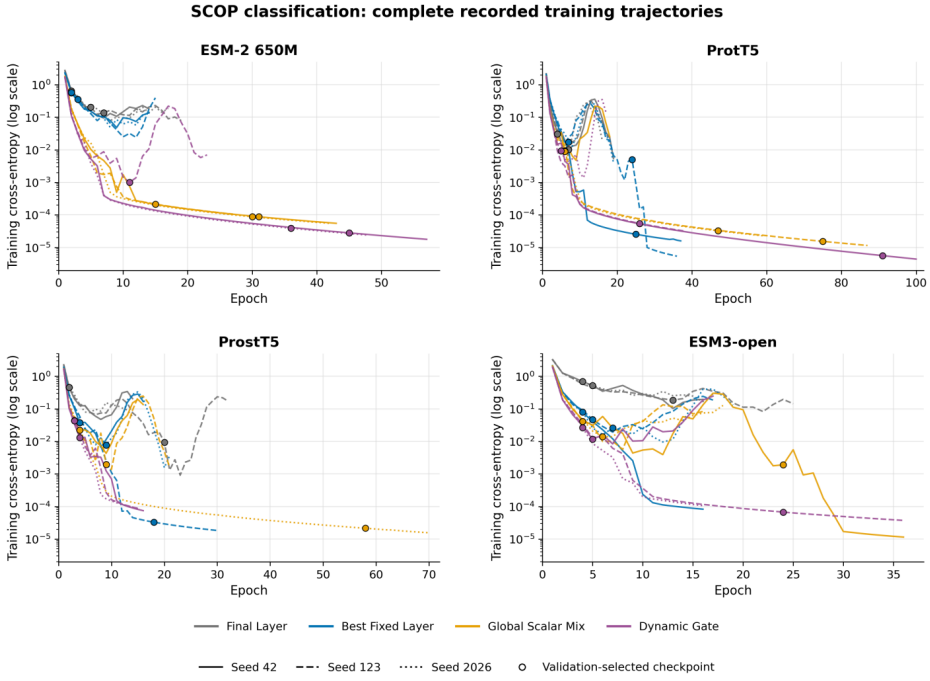


Figure C.1: Training cross-entropy for four backbone models on SCOP. Each curve represents a run of one method with one random seed, the logarithmic loss axis makes differences during later training stages more apparent. Markers indicate checkpoints selected based on validation accuracy. Final Layer and Best Fixed Layer train only their classifiers.

D CONTROLS TRAINED FROM SCRATCH AND LAYER WEIGHT ANALYSIS

Using ESM-2 650M as a representative model, we examine **layer weight constraints, additional trainable transformations, representation scale, similarity combinations and weight matching across proteins during training.** All interventions keep the pretrained encoder frozen. Post-training weight permutations and swaps do not update gate or predictor parameters. Because the supplementary experiments are independently trained under their respective settings, differences in absolute performance values may occur across experiments due to training variability.

D.1 MEAN-GATE AND SPARSE TOP-K GATE TRAINED FROM SCRATCH

For the ablation experiments, all variants, including Dynamic Gate, are independently trained from scratch under the corresponding experimental settings. Mean-Gate is trained from random initialization. At each training step, we first generate layer weights for each training protein used in that step, then average these weight vectors, use the same shared layer weights to fuse representations for all proteins, and update the gate based on the task loss. After training, mean layer weights are computed using a fixed set of training proteins, and these weights are held fixed for validation or testing. This parameterization differs from directly optimizing a set of layer scalars in Global Scalar Mix.

From the first training step, Top-k Gate selects the k highest-scoring layers for each protein, using only the normalized weights of these layers for fusion. We report results for k=4 and k=8. This experiment evaluates the network's performance when relearning under sparsity constraints.

We examine this behavior across four representative models, covering the Sequence-only and Structure-aware categories, including ESM-2 650M, ESM3-open, ProtT5, and ProstT5. The results are reported in Table D.2.

Table D.1: Performance on PISCES with layer weight constraints applied during training from scratch.

Method	**Spearman** $\uparrow$	**RMSE** $\downarrow$
Mean-Gate	0.6624 ± 0.0000	0.1367 ± 0.0000
Top-4 Gate	0.6036 ± 0.0395	0.1384 ± 0.0041
Top-8 Gate	$\mathbf{0.7334 \pm 0.0126}$	0.1322 ± 0.0004
Dynamic Gate	0.7211 ± 0.0133	$\mathbf{0.1287 \pm 0.0023}$

On PISCES, the full Dynamic Gate achieves lower RMSE than Mean-Gate and both sparse gates, but Top-8 has a slightly higher mean Spearman correlation than the full gate. On SCOP, Mean-Gate achieves slightly higher mean accuracy than the full gate on all three test sets, whereas Top-4 and Top-8 have lower mean accuracy. Therefore, the effects of using shared or sparse weights after retraining depend on the task and evaluation metric, **and full dynamic gating is not optimal in every setting. Yet we also find Mean-Gate to be a strong baseline, and one possible explanation is that for a given set of proteins or task, using only mean weights learned anew for all layers under task supervision is already sufficient.**

D.2 FINAL-LAYER TRANSFORMATION AND ALL-LAYER CONCATENATION CONTROLS

We introduce two types of MLP controls, targeting additional trainable parameter counts close to that of Dynamic Gate to control model size. Final Layer + MLP maps the final-layer protein vector to a new representation. All Layers + MLP concatenates all layer vectors and applies a learnable transformation. Both MLP controls receive supervision from the corresponding task, **and are used to assess the effects of additional nonlinear transformations and the direct joint use of information from all layers.**

Dynamic Gate achieves higher accuracy than both types of MLP controls on all three test sets for SCOP classification. **This result indicates that, compared with standard nonlinear transforma-**

Table D.2: Performance on SCOP with layer weight constraints applied during training from scratch Accuracy in percent.

Model	Method	Family ↑	Superfamily ↑	Fold ↑
ESM-2 650M	Dynamic Gate	98.98 ± 0.16	78.73 ± 1.18	36.21 ± 1.24
	Mean-Gate	**99.00 ± 0.05**	**78.76 ± 0.67**	**36.72 ± 1.01**
	Top-4 Gate	98.51 ± 0.24	72.94 ± 0.85	31.34 ± 1.25
	Top-8 Gate	98.40 ± 0.40	76.05 ± 0.99	32.64 ± 2.37
ESM3-open	Dynamic Gate	98.90 ± 0.08	78.10 ± 0.93	**37.98 ± 1.29**
	Mean-Gate	**98.98 ± 0.08**	**79.37 ± 0.26**	37.65 ± 1.68
	Top-4 Gate	97.93 ± 0.12	71.69 ± 2.52	34.82 ± 2.16
	Top-8 Gate	98.51 ± 0.31	74.93 ± 1.80	36.44 ± 1.34
ProtT5	Dynamic Gate	**99.34 ± 0.25**	**73.37 ± 1.48**	30.97 ± 1.03
	Mean-Gate	**99.34 ± 0.12**	70.39 ± 1.48	**31.29 ± 0.35**
	Top-4 Gate	98.30 ± 0.24	62.65 ± 0.39	27.44 ± 1.58
	Top-8 Gate	98.98 ± 0.34	67.92 ± 1.06	28.51 ± 1.85
ProstT5	Dynamic Gate	**99.24 ± 0.09**	75.12 ± 1.92	33.66 ± 1.48
	Mean-Gate	99.21 ± 0.08	**77.01 ± 2.08**	**34.08 ± 0.82**
	Top-4 Gate	98.24 ± 0.25	68.16 ± 0.99	31.06 ± 1.19
	Top-8 Gate	98.72 ± 0.25	70.84 ± 0.49	31.66 ± 1.60

Table D.3: Results for MLP controls and Dynamic Gate SCOP accuracy is reported in percent.

Datasets and Metrics	Final Layer + MLP	All Layers + MLP	Dynamic Gate
SCOP Family ↑	96.93 ± 0.28	96.78 ± 0.31	**98.98 ± 0.16**
SCOP Superfamily ↑	61.86 ± 0.80	61.27 ± 0.57	**78.73 ± 1.18**
SCOP Fold ↑	27.11 ± 0.95	28.69 ± 0.37	**36.21 ± 1.24**

tions of fixed representations, dynamically adjusting layer weights can yield greater improvements in downstream performance.

D.3 Random Permutation of Layer Weights Across Proteins

Without retraining the model, we randomly permute entire layer-weight vectors across test proteins. Each protein retains its own layerwise representations, with only the fusion coefficients replaced by the weight vector assigned through permutation. Layer indices within each weight vector remain unchanged. We use three random seeds for training, with 100 permutations per seed.

Permutation on PISCES is restricted to the 285 proteins involved in the 460 retained test pairs, and evaluation uses the same 460 test pairs. SCOP permutations are performed separately within the Family, Superfamily and Fold test sets, containing 1,272, 1,254 and 718 proteins respectively.

Permutation reduces the mean Spearman correlation on PISCES by 0.5016. SCOP accuracy decreases by 0.64, 1.76 and 1.22 percentage points respectively. The two task types use different metrics, so we do not directly compare the numerical magnitudes, but permutation clearly disrupts ranking quality on PISCES, and after permutation, ranking quality on PISCES declines markedly, whereas classification accuracy on SCOP decreases only slightly. This difference may relate to how task objectives and evaluation metrics respond to changes in representations: on PISCES, changes in fused representations may alter the similarity rankings of protein pairs; on SCOP, even if fused representations change, as long as samples do not cross a classification decision boundary, their predicted classes may remain unchanged.

Table D.4: Comparison of original and randomly permuted weights SCOP accuracy is reported in percent.

Dataset and Metric	Own Weights	Permuted Weights
PISCES Spearman ↑	**0.7211 ± 0.0133**	0.2195 ± 0.0048
PISCES RMSE ↓	**0.1288 ± 0.0024**	0.1625 ± 0.0004
SCOP Family ↑	**98.98 ± 0.16**	98.34 ± 0.05
SCOP Superfamily ↑	**78.73 ± 1.18**	76.97 ± 0.71
SCOP Fold ↑	**36.21 ± 1.24**	34.99 ± 0.71

Therefore, we suggest one possible explanation, namely that weight permutation perturbs relative relationships within representations, but this change is not fully reflected in classification accuracy. The smaller decrease in SCOP accuracy does not directly imply that protein-specific weights are ineffective, nor does it justify concluding that fused representations change only slightly. **As these results suggest, the role of input-dependent layer-weight matching is task-dependent, and its effects should be interpreted in the context of the specific task objectives and evaluation metrics.**

D.4 Weight Swaps Between Proteins of the Same or Different Classes

Using the discrete Fold labels in SCOP, we construct separate within-class and between-class weight swap conditions. Each target protein always retains its own layerwise representations, using only the full layer-weight vector of a protein from the same or a different class for fusion. For each of the three random seeds used for training, we perform 100 within-class swaps and 100 between-class swaps, without retraining the model. Class labels are used to construct this post hoc intervention condition.

Table D.5: Results of within-class and between-class weight swaps on SCOP Accuracy in percent.

Test Split	Own Weights	Within-Class Swap	Across-Class Swap
Family	**98.96 ± 0.16**	98.91 ± 0.14	98.38 ± 0.03
Superfamily	**82.21 ± 1.45**	81.75 ± 1.08	80.49 ± 0.92
Fold	**37.89 ± 1.39**	37.42 ± 0.89	36.62 ± 0.73

The performance decline caused by within-class swaps confirms that the weights are specific to individual proteins, showing that even proteins of the same class cannot fully share a layer fusion strategy; between-class swaps cause a larger decline, revealing that these weights encode class-related discriminative information. Together, these findings indicate that matching input-dependent layer weights to the target protein is effective for its downstream task. Notably, the decreases in classification accuracy caused by weight swaps are generally small. With the encoder frozen, and the target protein's layerwise representations unchanged, swapping alters only how these representations are combined, while changes in the fused representation do not necessarily alter the predicted class. **Although the absolute decreases are limited, the mean results on all three test sets consistently show larger decreases from between-class swaps than from within-class swaps, suggesting that layer weights contain class-related information, and providing additional evidence that matching weights to the target protein aids downstream prediction.**

E Experimental Details and Complete Results for Structure-Aware and Multimodal Models

E.1 Tasks and Inputs

We use S-PLM V1 and ProTrek-650M as two enhanced models based on the pretrained ESM-2 650M model, both of which take only amino acid sequences as input, to evaluate different representation readout strategies with their pretrained parameters frozen. This section details the SCOP classification experiments; comparative results for PISCES, SCOP and the CATH/EC tasks on shared proteins are presented in Sections E.5, E.4 and E.6 respectively. The SCOP experiments use the fixed splits and 1,195-class fold labels from Appendix A.

E.2 Pretrained Backbones and Feature Extraction

Table E.1: Implementation details for each backbone model.

Setting	S-PLM V1	ProTrek-650M
Pretrained Weights	Official S-PLM V1 sequence branch.	Sequence branch of the official `westlake-repl/ProTrek_650M` model.
Encoder	ESM-2 650M with pretrained structural adapters in layers 18–33.	ESM-2 650M sequence encoder trained by ProTrek.
Layer Features	Layers 1–33; 1,280 dimensions per layer.	Layers 1–33; 1,280 dimensions per layer.
Protein-Level Pooling	Following the upstream Fold implementation, mean pooling over all 512 positions, including BOS, EOS, and padding positions.	Hidden state of the first token (CLS) at each layer, with padding positions masked by the attention mask.
Processing After Layer Selection or Fusion	No additional projection, L2 normalization, or dataset-level standardization.	Frozen native linear projection ($1{,}280 \rightarrow 1{,}024$ dimensions, with bias), followed by L2 normalization.
Newly Added Classification Head	Linear layer ($1{,}280 \rightarrow 1{,}195$), with bias.	Linear layer ($1{,}024 \rightarrow 1{,}195$), with bias.

All pretrained parameters remain frozen, including the S-PLM adapters and the ProTrek projection layer. The encoders operate in evaluation mode, without using the structure or text branches. Layer 0 is excluded, and the output after the encoder's final LayerNorm is used for layer 33. Feature extraction uses FP16 automatic mixed precision, and features are cached in FP16. In both sets of comparisons, ProTrek's projection and subsequent L2 normalization are computed in FP32.

E.3 Representation Readout and Classifier

We compare Final Layer, Best Fixed Layer, Uniform Mean, Global Scalar Mix and Dynamic Gate. Final Layer uses layer 33, Best Fixed Layer uses the selected layer consistently within the same task, Uniform Mean averages layers 1–33 with equal weights, Global Scalar Mix learns layer weights shared across all inputs, and Dynamic Gate generates input-dependent weights for each protein. Following the model-specific processing in Table E.1, the resulting representations are fed into a single-layer linear classification head.

The gate first applies LayerNorm without affine parameters to each layer vector for scoring, then sequentially applies Linear(1,280, 64), tanh and a bias-free Linear(64, 1). Applying softmax to the 33 scores yields the layer weights. The weighted sum uses the original layer vectors, rather than the scoring inputs transformed by LayerNorm. The weights of the final scoring layer are initialized to zero, yielding uniform initial layer weights.

Fixed-layer readout and uniform averaging train only the newly added classifier, Global Scalar Mix jointly trains the layer weights and classifier, and Dynamic Gate jointly trains the gate and classifier from scratch, without transferring the PISCES gate.

E.4 Complete Results for the Three SCOP Test Splits

Table E.2: SCOP classification results for S-PLM and ProTrek-650M Accuracy in percent.

Model	Readout Method	Family ↑	Superfamily ↑	Fold ↑
S-PLM	Final Layer	**98.40 ± 0.09**	63.37 ± 0.32	**35.65 ± 0.50**
	Best Fixed Layer (L23)	97.80 ± 0.28	72.59 ± 1.08	32.36 ± 1.16
	Uniform Mean	98.14 ± 0.20	72.94 ± 1.15	33.98 ± 0.48
	Global Scalar Mix	98.06 ± 0.35	70.39 ± 3.02	34.49 ± 0.93
	Dynamic Gate	98.35 ± 0.31	**73.42 ± 1.43**	33.43 ± 2.19
ProTrek-650M	Final Layer	97.01 ± 0.14	48.27 ± 0.12	27.16 ± 0.14
	Best Fixed Layer (L21)	**98.43 ± 0.00**	**64.97 ± 0.09**	**34.35 ± 0.08**
	Uniform Mean	98.14 ± 0.05	61.35 ± 0.17	33.70 ± 0.14
	Global Scalar Mix	98.35 ± 0.08	64.11 ± 0.70	34.31 ± 0.43
	Dynamic Gate	97.06 ± 0.20	63.85 ± 0.12	28.37 ± 0.56

Dynamic Gate with S-PLM achieves the highest mean accuracy on Superfamily and improves upon final-layer readout by 10.05 percentage points, but Final Layer achieves higher accuracy on Family and Fold. The best fixed layer for ProTrek-650M yields higher mean accuracy than Dynamic Gate on all three test sets. One possible explanation is as follows: enhanced models reorganize the distribution of task-relevant information across depths and the complementarity between layers.

E.5 PISCES Structural Similarity Results

Table E.3: PISCES results for S-PLM and ProTrek-650M.

Model	Readout Method	Spearman ↑	RMSE ↓
S-PLM	Final Layer	0.7397	0.1577
	Best Fixed Layer (L18)	0.6222	0.1408
	Uniform Mean	0.6725	0.1581
	Global Scalar Mix	0.6601 ± 0.0000	0.1482 ± 0.0000
	Dynamic Gate	**0.7453 ± 0.0038**	**0.1347 ± 0.0001**
ProTrek-650M	Final Layer	0.4175	0.1449
	Best Fixed Layer (L24)	0.5504	0.1361
	Uniform Mean	0.5620	0.1351
	Global Scalar Mix	0.5509 ± 0.0000	0.1332 ± 0.0000
	Dynamic Gate	**0.7212 ± 0.0184**	**0.1117 ± 0.0042**

Dynamic Gate achieves the highest mean Spearman correlation and the lowest RMSE for both models. The final layer of S-PLM already exhibits high rank correlation, so dynamic fusion yields a relatively small increase in Spearman correlation over final-layer readout, whereas ProTrek-650M shows a more pronounced improvement. These results further show that Dynamic Gate improves structural similarity readout for both S-PLM and ProTrek-650M. Whether for S-PLM with its high initial rank correlation, or for ProTrek-650M with greater room for improvement, our method consistently ranks first. **This indicates that dynamic fusion can still provide additional gains when the final-layer representation is already strong, while the magnitude of the improvement varies across models.**

E.6 CATH AND EC RESULTS ON SHARED PROTEINS

Table E.4: CATH and EC classification results for S-PLM and ProTrek-650M Accuracy in percent

Model	Readout Method	CATH ↑	EC ↑
S-PLM	Final Layer	89.70 ± 1.60	67.47 ± 0.93
	Best Fixed Layer	92.93 ± 2.45	**73.33 ± 1.60**
	Global Scalar Mix	93.13 ± 0.93	69.70 ± 1.60
	Dynamic Gate	**94.95 ± 0.93**	68.28 ± 0.93
ProTrek-650M	Final Layer	80.00 ± 1.21	65.86 ± 0.93
	Best Fixed Layer	83.64 ± 2.64	**73.54 ± 1.26**
	Global Scalar Mix	86.67 ± 0.61	71.72 ± 2.80
	Dynamic Gate	**86.87 ± 2.29**	70.10 ± 0.93

However, we find that in SCOP classification, **after applying our method to the base model ESM-2 650M, the downstream performance of the fused representations exceeds not only that of the reproduced S-PLM final-layer representations, but also that obtained by applying the same fusion method to S-PLM.** This observation suggests that, despite an enhanced model's advantage under final-layer readout, its full layerwise representation space does not necessarily offer greater downstream utility. Furthermore, as a supplementary reference, the original S-PLM paper reports accuracies of 98.36%, 62.50% and 34.72% on the Family, Superfamily and Fold test sets respectively, which are closely consistent with our reproduced final-layer results of 98.40%, 63.37% and 35.65%. Notably, although S-PLM enhances ESM-2 650M through structure-aware pretraining, applying our multilayer readout method directly to the original ESM-2 650M, without modifying its pretrained encoder, achieves accuracies of 98.98%, 79.21% and 37.05% on the three test sets, respectively, outperforming S-PLM across all three settings. **This comparison shows that encoder enhancement is not the only route to improving downstream performance, and that more effective utilization of existing layerwise representations can provide an independent source of gains without modifying the pretrained encoder.**

Overall, the downstream performance of protein language models is task-dependent: different models may exhibit different strengths in tasks such as structural similarity evaluation, structural classification and functional classification, and their relative performance is also influenced by the representation readout strategy. Therefore, when evaluation is limited to a single task or layer, the results may not fully reflect the downstream utility of a model's existing representations. Meanwhile, between PLM representation dimensions and protein structure, function and other biological attributes, there is not necessarily a direct and fixed one-to-one correspondence (Tule et al., 2025), and relevant information may be distributed across multiple dimensions, different layers and their combinations. Although existing feature analysis methods can identify some representation features associated with protein properties, the stability, biological meaning and scope of applicability of these associations require further clarification. **Against this background, we focus on the readout and use of layerwise representations, examining how existing information can be reorganized under target-task supervision, providing a complementary perspective on the relationship between pretrained representations and downstream performance, and offering directions for further research on and use of protein language model representations.**