PROBREAD: ALLOSTERIC READOUT AND HOMEOSTATIC FUSION FOR FROZEN CROSS-MODAL BIOMOLECULAR MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-modal biomolecular models for molecules and proteins use an encoder for each modality and a projector that maps the pooled encoder output into a shared space. For downstream tasks, these models are typically frozen and evaluated by linear probing on a single representation: the pooled encoder embedding or the projected embedding. We observe that the better representation changes across tasks. Moreover, fixed pooling applies the same task-independent rule to every input, and concatenating the two pooled embeddings yields limited gains. We propose **ProbRead** (**Prob**e **Read**out), a lightweight readout with two components: (i) Allosteric Readout uses the projected embedding only to guide token selection over the encoder tokens, replacing fixed pooling. (ii) Homeostatic Fusion assigns complementary, input-conditioned weights to the aggregated token representation and the projected embedding in each feature dimension. Across three platforms and five groups of biomolecular tasks, ProbRead achieves the best average performance, outperforming encoder-only and projector-only probing as well as linear probing of other biomolecular foundation models. ProbRead's output also recovers the effective rank lost by projection. These results demonstrate that adaptively combining the two representations is more effective than fixing either one.

1 INTRODUCTION

Cross-modal contrastive learning aligns molecular structures with text [1; 2], protein sequences with 3D structures [3], and images with text [4; 5]. In these models, each modality has an encoder that produces token embeddings, and a projection head maps the pooled encoder output into a shared space where the contrastive loss is applied. Because labeled data are often limited and full fine-tuning is expensive, downstream applications typically freeze the pretrained encoder and projection head. The frozen model thus provides two representations: the encoder token embeddings and the projected embedding. Standard linear probing trains a linear predictor on a single representation: either the encoder tokens aggregated by pooling [6; 7], or the projected embedding [4; 8]. This readout is simple but has two limitations: fixed pooling applies the same task-independent rule to every input, and the choice of representation is fixed rather than adapted to the task.

The choice of representation matters because the better representation can change across tasks, even within the same frozen model. Figure 1 shows this reversal for ProteinCLIP tasks [8]. The projected embedding outperforms the pooled encoder embedding on protein–protein interaction (PPI) prediction (66.72 vs. 61.95 AUC) but underperforms it on CATH structural classification (66.49 vs. 68.81 accuracy). Similar reversals appear across the molecular property prediction tasks (Tables 2–4 in Section 4.3). These results suggest that the two representations make different task-relevant features accessible to a linear predictor. Fixing either representation in advance can therefore be suboptimal.

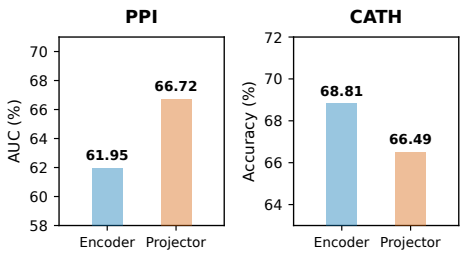

Figure 1: The better representation changes across tasks (PPI and CATH) with the same frozen model.

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.

The reversal reflects complementary strengths of the two embeddings. Encoder token embeddings capture fine-grained, modality-specific features such as functional residues and molecular substructures [9; 10]. Fixed pooling, however, aggregates them with a task-independent rule and suppresses token-level information. Projected embeddings are optimized for cross-modal alignment and encode modality-shared semantics in a compact space, making them easier to separate with a linear classifier. This compression removes modality-specific features needed by downstream tasks [11; 12]. Prior work reports stronger transfer from encoder features before projection [13; 14], but the reversals in Figure 1 and Tables 2–4 show that projected embeddings remain preferable for some tasks.

A simple way to use both representations is to concatenate the pooled encoder embedding with the projected embedding. In our experiments, this yields limited and inconsistent gains (Table 1). Concatenation also has two structural limitations. First, the projected embedding is computed from the pooled encoder embedding, so concatenation cannot recover token-level information lost in pooling. Second, a linear predictor applies the same weights to every input, so the balance between the two representations cannot vary across inputs. An effective readout should perform task-adaptive token selection over the encoder tokens and input-conditioned balancing between the two representations.

We propose ProbRead, short for Probe Readout, a lightweight readout for frozen cross-modal models. It contains two components that address the two requirements above in turn. (i) Allosteric Readout (AR) is inspired by allosteric regulation, in which a regulator binds at a site distinct from the active site and modulates activity there without being the substrate. AR uses the projected embedding as the query of a cross-attention over the encoder token embeddings. Like such a regulator, the projected embedding determines which tokens to read but does not enter the output, which is computed from the token embeddings alone. AR thus replaces fixed pooling with task-adaptive, input-conditioned token selection. (ii) Homeostatic Fusion (HF) is inspired by homeostasis, in which a system keeps a balance between opposing processes and shifts it as conditions change. HF combines the AR output with the projected embedding through a gate conditioned on both. In each feature dimension, it assigns complementary weights to the two representations, so raising one lowers the other. The gate adapts this balance to each input rather than fixing it for all inputs.

Our contributions are: **(i)** We identify a task-dependent trade-off between encoder and projector representations in frozen cross-modal models. Performance reversals, effective-rank analysis, and concatenation experiments show that neither representation provides a universally effective readout. **(ii)** We propose ProbRead, a lightweight Probe Readout that introduces Allosteric Readout for task-adaptive information selection and Homeostatic Fusion for adaptive information balancing. The two components address the limitations of fixed pooling and static information combination, respectively. **(iii)** We evaluate ProbRead on three cross-modal platforms covering five groups of biomolecular tasks. With all backbones frozen, ProbRead achieves the best average performance in every task group, outperforming encoder-only and projector-only probing as well as linear probing of other pretrained models. On the analyzed datasets, its output keeps 81–123% of the encoder's effective rank, indicating that the readout does not introduce substantial dimensional collapse.

2 Related Work

Cross-modal learning. Cross-modal contrastive learning [15; 6; 16; 4] is widely used to align representations across modalities, including molecular [17; 2; 1; 18; 19], protein [3; 7; 20; 21; 22], and vision–language [4] domains. These methods learn a shared embedding space via a projection head that maps each encoder's output into the space where the contrastive loss operates. A key design choice lies in determining the role of the projection head during evaluation. One approach, adopted by CLIP [4], TRIDENT [1], and ProteinCLIP [8], uses the projector output as the downstream representation. This preserves shared information but may reduce access to unique features [11; 12]. Another approach, originating from Chen et al. [6] and followed by 3D Infomax [18], MoleculeSTM [2], ProtST [7], and ProTrek [20], discards the projector to avoid this loss but may not fully utilize the shared information retained by the projector. However, neither approach explicitly regulates the use of these complementary information sources. Projection heads therefore create a trade-off between shared and modality-unique information, motivating readout strategies that can selectively access and balance information after pretraining.

Biomolecular foundation models. Self-supervised pretraining has produced diverse biomolecular foundation models that learn rich token-level representations. In the molecular domain, SMILES-

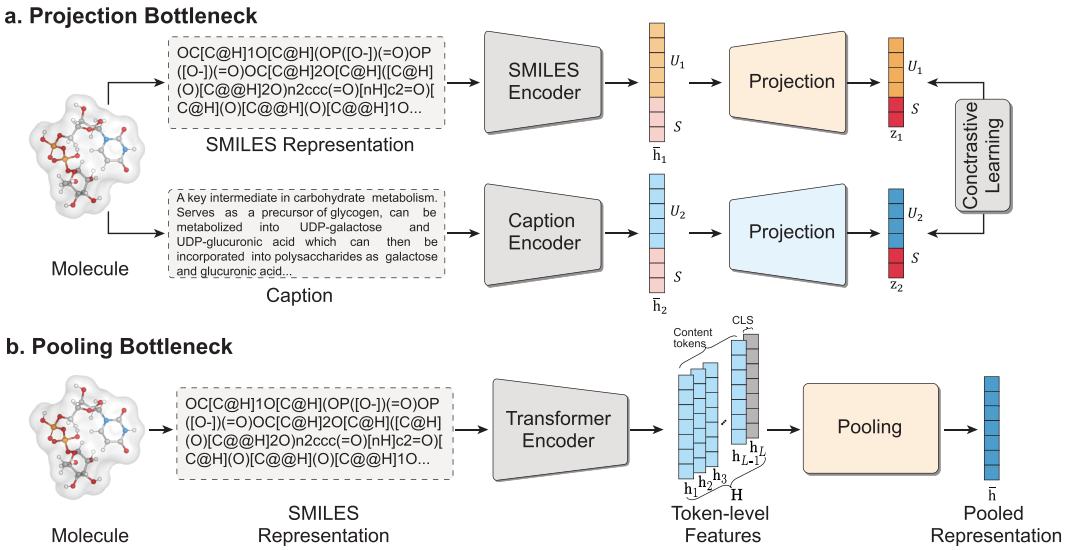

Figure 2: The dual bottleneck in cross-modal contrastive learning. (a) Projection bottleneck. Two modalities are encoded and projected into a shared space. The encoder output $\bar{\mathbf{h}}_i$ carries both modality-specific U_i and shared S; the projector output Z_i retains only S, discarding some modality-specific features. (b) Pooling bottleneck. Fixed pooling (CLS or mean) compresses the token sequence into a single vector $\bar{\mathbf{h}}$, discarding positional and local structure.

based language models such as MoLFormer [10], ChemBERTa [23], SELFormer [24], SMI-TED [25], and ChemFM [26] encode local chemical context through token representations, including functional groups, ring systems, and stereochemistry, whereas graph neural networks [27; 28] and 3D-aware models such as Uni-Mol [29] capture molecular topology and spatial structure at the atom level. In the protein domain, sequence language models such as ESM-2 [30], ESM-C [31], and ProtTrans [32] yield residue-level embeddings that capture secondary structure, binding sites, and contact maps without explicit structural supervision [33; 9]. Structure-aware extensions such as SaProt [34] and GearNet [21] enrich residue-level features with geometric information, whereas cross-modal variants including ProtST [7] and ProTrek [20] align sequences with text or structure. Despite this token-level richness, standard downstream evaluation reduces the sequence to a single vector via CLS-token selection or mean pooling before linear probing. Task-relevant local patterns, such as catalytic residues or reactive functional groups, may be diluted or lost during this compression. Moreover, downstream probing typically selects one representation source, leaving the regulation of complementary information sources underexplored. See more related work about projection heads and token aggregation in Appendix A.

3 Method

3.1 Problem Formulation

We consider a cross-modal model pretrained to align N modalities. We present the formulation for two modalities; the extension to more modalities is straightforward. For modality $m \in \{1, 2\}$, an encoder $f_{\boldsymbol{\theta}_m}$ with parameters $\boldsymbol{\theta}_m$ maps an input $\mathbf{x}_m$ to a sequence of token embeddings: $\mathbf{H}_m = f_{\boldsymbol{\theta}_m}(\mathbf{x}_m) = [\mathbf{h}_{m,1}, \ldots, \mathbf{h}_{m,L_m}] \in \mathbb{R}^{L_m \times d_e}$, where L_m is the number of tokens and d_e is the encoder dimension. A fixed pooling operator P_m, such as mean or CLS pooling, produces a single embedding $\bar{\mathbf{h}}_m = P_m(\mathbf{H}_m) \in \mathbb{R}^{d_e}$. A projection head $g_{\boldsymbol{\phi}_m}$ with parameters $\boldsymbol{\phi}_m$ maps this embedding into a d_p-dimensional shared space: $\mathbf{z}_m = g_{\boldsymbol{\phi}_m}(\bar{\mathbf{h}}_m) \in \mathbb{R}^{d_p}$, where d_p is the projector dimension. The encoder and projector are frozen for downstream tasks.

For a downstream task on modality m, let $\mathcal{D}_m = \{(\mathbf{x}_m^{(i)}, y^{(i)})\}_{i=1}^{n_m}$ denote a labeled dataset with n_m examples. Standard linear probing fits a linear predictor to either the pooled encoder embedding $\bar{\mathbf{h}}_m$ or the projected embedding $\mathbf{z}_m$. These choices expose two information bottlenecks in the representation pipeline, which maps $\mathbf{H}_m$ to $\bar{\mathbf{h}}_m$ and then to $\mathbf{z}_m$. Let $\mathcal{I}(A; B)$, $\mathcal{I}(A; B \mid C)$, and

$\mathcal{I}(A; B; C)$ denote mutual information, conditional mutual information, and interaction information, respectively. All information quantities are defined over the data distribution.

Projection bottleneck. We define the task-relevant shared information as $S \triangleq \mathcal{I}(\mathbf{x}_1; \mathbf{x}_2; y)$. The modality-specific information is $U_1 \triangleq \mathcal{I}(\mathbf{x}_1; y \mid \mathbf{x}_2)$ for modality 1 and $U_2 \triangleq \mathcal{I}(\mathbf{x}_2; y \mid \mathbf{x}_1)$ for modality 2. The total task-relevant information satisfies $\mathcal{I}(\mathbf{x}_1, \mathbf{x}_2; y) = S + U_1 + U_2$. Contrastive learning rewards information shared across modalities but provides no explicit objective for preserving U_1 or U_2. Under the minimal shared-representation assumption [11], the projected embeddings preserve shared information but suppress task-relevant modality-specific information. Since $\mathbf{z}_m = g_{\phi_m}(\bar{\mathbf{h}}_m)$ is a deterministic function of $\bar{\mathbf{h}}_m$, the chain rule gives

$$\mathcal{I}(\bar{\mathbf{h}}_m; y) = \mathcal{I}(\mathbf{z}_m; y) + \underbrace{\mathcal{I}(\bar{\mathbf{h}}_m; y \mid \mathbf{z}_m)}_{\text{projection gap}} \geq \mathcal{I}(\mathbf{z}_m; y). \tag{1}$$

The projection gap quantifies task-relevant information contained in the pooled encoder embedding but not captured by the projected embedding. This ordering does not determine linear-probing performance. The projector may improve linear separability by organizing shared information into a compact embedding space. Using $\mathbf{z}_m$ alone may suppress modality-specific information, whereas discarding it removes the cross-modal structure learned during pretraining.

Pooling bottleneck. The fixed operator P_m compresses the token sequence $\mathbf{H}_m \in \mathbb{R}^{L_m \times d_e}$ into a single embedding $\bar{\mathbf{h}}_m \in \mathbb{R}^{d_e}$. Since $\bar{\mathbf{h}}_m = P_m(\mathbf{H}_m)$ is also deterministic,

$$\mathcal{I}(\mathbf{H}_m; y) = \mathcal{I}(\bar{\mathbf{h}}_m; y) + \underbrace{\mathcal{I}(\mathbf{H}_m; y \mid \bar{\mathbf{h}}_m)}_{\text{pooling gap}} \geq \mathcal{I}(\bar{\mathbf{h}}_m; y). \tag{2}$$

The pooling gap quantifies task-relevant token-level information not captured by fixed pooling. Since $\mathbf{z}_m$ is computed from $\bar{\mathbf{h}}_m$, $\mathcal{I}(\bar{\mathbf{h}}_m, \mathbf{z}_m; y) = \mathcal{I}(\bar{\mathbf{h}}_m; y)$, thus combining the two pooled embeddings cannot recover the information $\mathcal{I}(\mathbf{H}_m; y \mid \bar{\mathbf{h}}_m)$. Combining Equations (1) and (2) yields

$$\mathcal{I}(\mathbf{H}_m; y) - \mathcal{I}(\mathbf{z}_m; y) = \underbrace{\mathcal{I}(\mathbf{H}_m; y \mid \bar{\mathbf{h}}_m)}_{\text{pooling gap}} + \underbrace{\mathcal{I}(\bar{\mathbf{h}}_m; y \mid \mathbf{z}_m)}_{\text{projection gap}}. \tag{3}$$

This decomposition gives two design requirements. First, the downstream readout should aggregate token embeddings directly from $\mathbf{H}_m$ using task supervision. Second, it should adaptively fuse the resulting token-level representation with the projected embedding $\mathbf{z}_m$. Next, we propose ProbRead to satisfy both requirements.

3.2 The Proposed ProbRead Framework

ProbRead consists of Allosteric Readout (AR) and Homeostatic Fusion (HF). AR performs projector-conditioned token aggregation, and HF adaptively fuses the resulting representation with the projector features. Together, they address the pooling and projection bottlenecks with the encoder and projector frozen. Figure 3 shows the architecture, and Algorithm 1 in Appendix B summarizes the procedure.

3.2.1 Allosteric Readout (AR)

Fixed pooling applies a predefined aggregation rule to every input. Mean pooling assigns uniform weights to all tokens, whereas CLS pooling always selects the designated CLS token. Neither operation uses downstream supervision to determine which tokens are relevant. As characterized by the pooling gap in Equation (3), task-relevant token features may not be accessible from the pooled representation. An effective readout should access $\mathbf{H}_m$ directly, learn task-specific token importance, and adapt its selection to each input.

Allosteric design principle. Our design is inspired by allosteric regulation, where a regulatory signal modulates activity at another site without becoming the content being processed. Following this principle, AR separates *regulation* from *content*. The projector representation $\mathbf{z}_m$ serves as a global regulatory signal that controls token selection, and the encoder sequence $\mathbf{H}_m$ provides the content to be aggregated. We implement this interaction by single-query multi-head cross-attention.

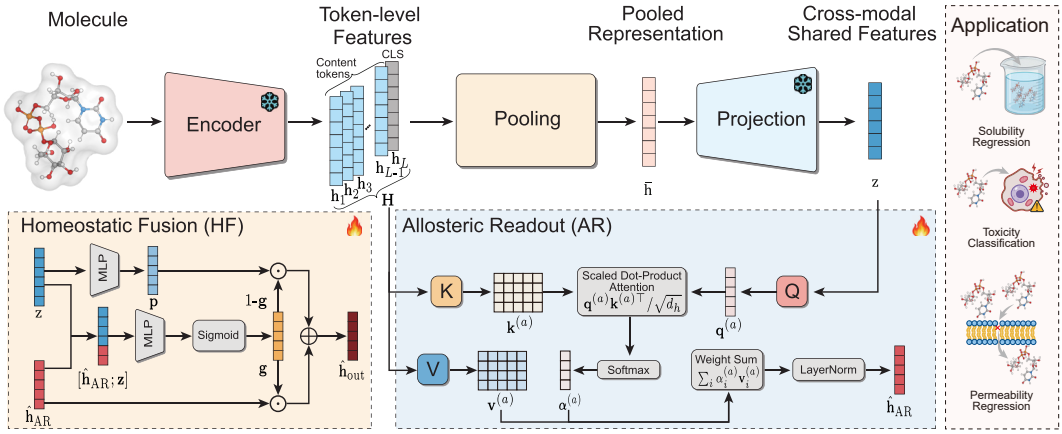

Figure 3: Overview of ProbRead. The frozen encoder and projector provide fine-grained token representations and a cross-modal alignment signal, respectively. The trainable Allosteric Readout module leverages the projector signal to guide task-adaptive token selection from the encoder. Then, the Homeostatic Fusion module adaptively balances the retrieved representation and projector signal for downstream prediction. Frozen components are shown in blue and trainable components in red.

Projector-conditioned token scoring. AR uses n_h attention heads. Let $d_h = d_e/n_h$ denote the dimension of each head. For head $a \in \{1, \ldots, n_h\}$, let $\mathbf{W}_Q^{(a)} \in \mathbb{R}^{d_h \times d_p}$ denote the query projection, and let $\mathbf{W}_K^{(a)}, \mathbf{W}_V^{(a)} \in \mathbb{R}^{d_h \times d_e}$ denote the key and value projections. We compute $\mathbf{q}_m^{(a)} = \mathbf{W}_Q^{(a)} \mathbf{z}_m, \quad \mathbf{k}_{m,i}^{(a)} = \mathbf{W}_K^{(a)} \mathbf{h}_{m,i}, \quad \mathbf{v}_{m,i}^{(a)} = \mathbf{W}_V^{(a)} \mathbf{h}_{m,i}$. The query $\mathbf{q}_m^{(a)}$ carries the regulatory signal from the projector. The keys $\mathbf{k}_{m,i}^{(a)}$ represent the token features against which this signal is evaluated. The values $\mathbf{v}_{m,i}^{(a)}$ provide the content retrieved from each token. For head a, define the bilinear interaction matrix $\mathbf{A}^{(a)} = {\mathbf{W}_Q^{(a)}}^\top \mathbf{W}_K^{(a)} \in \mathbb{R}^{d_p \times d_e}$. The attention score between the projector representation and token i is $e_{m,i}^{(a)} = {\mathbf{q}_m^{(a)}}^\top \mathbf{k}_{m,i}^{(a)} / \sqrt{d_h} = \mathbf{z}_m^\top \mathbf{A}^{(a)} \mathbf{h}_{m,i} / \sqrt{d_h}$. This bilinear interaction implements allosteric regulation: the projector signal modulates the scoring of each encoder token without replacing its content. Each head learns a task-specific compatibility function between the cross-modal projector space and the encoder token space.

Allosteric token aggregation. We normalize the scores into attention weights $\alpha_{m,i}^{(a)}$ and compute the output $\mathbf{o}_m^{(a)}$ of each head:

$$\alpha_{m,i}^{(a)} = \frac{\exp(e_{m,i}^{(a)})}{\sum_{j=1}^{L_m} \exp(e_{m,j}^{(a)})}, \qquad \mathbf{o}_m^{(a)} = \sum_{i=1}^{L_m} \alpha_{m,i}^{(a)} \mathbf{v}_{m,i}^{(a)}. \tag{4}$$

The attention weights encode the manner in which the projector regulates token selection. The head output retrieves the corresponding content from the encoder values. Different heads can capture distinct compatibility patterns between global cross-modal semantics and local token features.

Let $\mathbf{W}_O \in \mathbb{R}^{d_e \times d_e}$ denote the output projection. We concatenate the head outputs and obtain the AR representation: $\hat{\mathbf{h}}_{\mathrm{AR},m} = \mathrm{LayerNorm}\Big(\mathbf{W}_O[\mathbf{o}_m^{(1)}; \ldots; \mathbf{o}_m^{(n_h)}]\Big)$. Layer normalization stabilizes the scale of the retrieved representation before its subsequent fusion with the projector representation. The encoder and projector remain frozen. The attention parameters $\{\mathbf{W}_Q^{(a)}, \mathbf{W}_K^{(a)}, \mathbf{W}_V^{(a)}\}_{a=1}^{n_h}$ and $\mathbf{W}_O$ are optimized using downstream supervision.

AR generalizes fixed pooling through input-conditioned token selection. Uniform attention follows the aggregation pattern of mean pooling, whereas a concentrated attention distribution approximates single-token pooling. The projector controls *where* to read, and the encoder tokens determine *what* information is retrieved. This separation preserves the allosteric design principle throughout the readout. The projector representation affects AR only through the attention weights; it is not included directly in $\hat{\mathbf{h}}_{\mathrm{AR},m}$. AR therefore addresses the pooling bottleneck but does not explicitly

preserve the projector features in the final representation. This limitation motivates Homeostatic Fusion, which adaptively fuses the retrieved token representation with the projector representation.

3.2.2 HOMEOSTATIC FUSION (HF)

AR constructs $\hat{\mathbf{h}}_{\mathrm{AR},m}$ from the token values in $\mathbf{H}_m$. The projector representation $\mathbf{z}_m$ regulates token selection but does not enter the AR output directly. AR therefore provides direct access to token-level features but retains the cross-modal projector features only indirectly. Discarding $\mathbf{z}_m$ would lose a representation that can remain useful for downstream prediction.

Simple fusion strategies do not fully address this limitation. Concatenation preserves both representations, but a linear prediction head applies input-independent fusion weights. A fixed sum assigns the same weight to every input and representation dimension. A scalar gate can adapt the overall mixture but cannot select different sources for different feature dimensions. An effective fusion mechanism should preserve a direct pathway from $\mathbf{z}_m$, align the two representations, and adjust their relative contributions across inputs and dimensions.

Homeostatic design principle. Our design is inspired by homeostasis, in which a system maintains a functional balance by adjusting interacting processes as conditions change. Following this principle, Homeostatic Fusion conditions its decision on both the AR and projector representations. It assigns complementary weights to the two pathways, so increasing the contribution of one pathway decreases the contribution of the other. The balance varies across inputs and feature dimensions. This design preserves both information sources without fixing the fusion rule globally.

Input-conditioned dimension-wise fusion. Since the AR representation $\hat{\mathbf{h}}_{\mathrm{AR},m} \in \mathbb{R}^{d_e}$ and projector representation $\mathbf{z}_m \in \mathbb{R}^{d_p}$ belong to different feature spaces, we first map $\mathbf{z}_m$ into the encoder dimension. Let $\mathbf{W}_p \in \mathbb{R}^{d_e \times d_p}$ denote a trainable projection matrix. The transformed projector representation is $\mathbf{p}_m = \mathbf{W}_p \mathbf{z}_m \in \mathbb{R}^{d_e}$. This transformation places the projector features and AR features in representations of the same dimension, enabling dimension-wise fusion.

Integration with AR and prediction. HF determines the balance from both information sources. Let $\mathbf{W}_g \in \mathbb{R}^{d_e \times (d_e + d_p)}$ denote a trainable gating matrix. We compute the balance vector $\mathbf{g}_m = \sigma\Big(\mathbf{W}_g[\hat{\mathbf{h}}_{\mathrm{AR},m}; \mathbf{z}_m]\Big) \in (0,1)^{d_e}$. Conditioning $\mathbf{g}_m$ on both representations allows HF to assess their joint feature state before assigning their contributions. Each element of $\mathbf{g}_m$ controls the balance for one output dimension. HF produces the final representation through

$$\hat{\mathbf{h}}_{\mathrm{out},m} = \mathbf{g}_m \odot \hat{\mathbf{h}}_{\mathrm{AR},m} + (\mathbf{1} - \mathbf{g}_m) \odot \mathbf{p}_m, \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^{d_e}$ denote the all-ones vector. The complementary coefficients $\mathbf{g}_m$ and $\mathbf{1} - \mathbf{g}_m$ implement the homeostatic design principle. They create a bounded trade-off between the token and projector pathways in every output dimension.

HF complements the regulatory role of AR. AR uses the projector representation to determine *where* to retrieve information from the encoder tokens. HF determines the amount of information retained from the retrieved token representation and the projector representation. AR bypasses fixed pooling, and HF replaces static feature fusion with an input-conditioned balance. Together, they provide direct access to $\mathbf{H}_m$ and preserve an adaptive pathway from $\mathbf{z}_m$.

Let $c_{\boldsymbol{\omega}}$ denote the task-specific prediction head with trainable parameters $\boldsymbol{\omega}$. Given the fused representation, the final prediction is $\hat{y} = c_{\boldsymbol{\omega}}(\hat{\mathbf{h}}_{\mathrm{out},m})$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Platforms and tasks. We evaluate three cross-modal contrastive models, referred to as platforms, each with a frozen encoder and projection head. TRIDENT [1] aligns molecules with text and taxonomic annotations using a MoLFormer-XL encoder [10]. S-PLM [3] aligns protein sequences with structures using an adapter-augmented ESM-2 650M encoder [30] . ProteinCLIP [8] aligns protein sequences with functional text using an ESM-2 150M encoder. On TRIDENT, we evaluate molecular property classification on 8 MoleculeNet [35] and 6 TDC [36] datasets (AUC) and regression on

4 datasets (RMSE). On S-PLM, we evaluate protein function prediction on EC and GO [37] (Fmax). On ProteinCLIP, we evaluate protein–protein interaction (PPI) prediction (AUC) and CATH [38] architecture classification (accuracy). These form the five task groups used throughout the paper. We further test a GNN encoder with MoleculeSTM [2] in Section 4.3.

Training. We freeze all encoders and projectors and train only ProbRead and the prediction head with AdamW [39] (learning rate and weight decay 10^{-4}, cosine annealing), selecting checkpoints on the validation set. Results are averaged over three seeds. Appendix C provides full configurations and data splits, and Appendix H defines the metrics.

Baselines. We compare ProbRead with a broad set of strong baselines covering different architectures and modalities. (i) Enc and Proj use the frozen: pooled encoder representation and projector output, respectively, to evaluate the benefit of jointly exploiting both representations over using either alone. (ii) For molecular property prediction, we include sequence-based language models ChemBERTa [23], SELFormer [24], MoLFormer-XL [10], and RoBERTa-ZINC [40], the 3D molecular model Uni-Mol [29], and the multimodal encoder-decoder BioT5+ [41]. (iii) For protein tasks, we include the large-scale sequence models ESM-2 [30] and ESM-C [31], the structure-aware model SaProt [34], and the cross-modal contrastive models ProtST [7] and ProTrek [20]. The sequence, structure-aware, and multimodal models establish strong performance references across major pretraining paradigms, whereas ProtST and ProTrek compare ProbRead against the standard practice of discarding the projector after cross-modal pretraining. ProbRead and all baselines use official pretrained weights under the same linear-probing protocol.

4.2 Dual Bottleneck Analysis

Before evaluating ProbRead, we examine the manifestation of the dual bottleneck identified in Section 3.1. We use effective rank (erank) [42] to verify that projection compresses representation geometry, and a concatenation diagnostic to confirm that encoder and projector representations carry distinct information yet cannot be effectively balanced through static combination. The erank quantifies the effective dimensionality of a representation: given a centered matrix $\mathbf{M} \in \mathbb{R}^{N\times d}$ with singular values σ_i, $\mathrm{erank}(\mathbf{M}) = \exp(-\sum_i p_i \log p_i)\,, \quad p_i = \frac{\sigma_i}{\sum_j \sigma_j}$. Garrido et al. [43] show that erank is a reliable unsupervised proxy for LP performance. We select datasets with $N > d$ so the upper bound is governed by dimensionality, not sample size.

Projection bottleneck analysis. Figure 4 compares encoder and projector erank across three platforms (see Appendix D for full data). The pattern is consistent: projection compresses representation diversity but retains cross-modal alignment information. TRIDENT's projector reduces erank from about 400 to 210; S-PLM's projector causes the most severe compression, with erank dropping from about 640 to 48 (up to 93% loss); ProteinCLIP drops from about 316 to 88. The severity correlates with the compression ratio of the projection head, confirming that the projection bottleneck identified in Section 3.1 manifests consistently across architectures and domains.

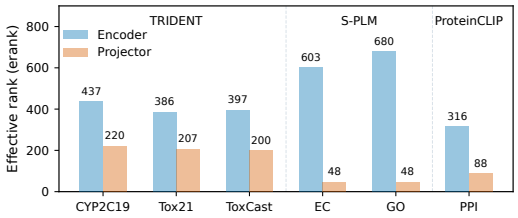

Figure 4: Effective rank of embeddings extracted by Encoder and projector. The projection bottleneck is consistent across all evaluated platforms.

Complementarity and limitation of pooled fusion. If encoder and projector representations carry complementary information, concatenation $[\bar{\mathbf{h}}; \mathbf{z}]$ should improve over either alone. Table 1 report the results on three platforms. On TRIDENT and S-PLM, concatenation indeed outperforms both individual representations, confirming that the two sources carry different strengths. However, on ProteinCLIP, where the projector representation carries stronger task-relevant features than the encoder representation, concatenation degrades performance, as the encoder's less relevant dimensions dilute the projector's compact signal.

Table 1: Results of Encoder, Projector, and their concatenation under linear-probing protocol.

	TRIDENT (cls)	TRIDENT (reg)	S-PLM (EC)	S-PLM (GO)	ProteinCLIP (PPI)
Enc	**77.49**	1.978	0.750	0.468	61.95
Proj	75.65	1.882	0.387	0.291	**66.72**
Concat	77.31	**1.720**	**0.752**	**0.470**	65.98

Table 2: AUC (↑) on eight MoleculeNet classification datasets (TRIDENT).

	Params	BACE	BBBP	ClinTox	HIV	MUV	SIDER	Tox21	ToxCast	Avg
BioT5+ [41]	252M	66.1	87.2	75.2	77.5	59.9	57.8	69.9	61.6	69.4
ChemBERTa [23]	77M	78.4	86.4	63.8	77.8	75.5	61.0	73.0	64.1	72.5
SELFormer [24]	85M	72.0	82.0	**89.5**	76.0	73.8	59.8	69.6	58.3	72.6
MoLFormer [10]	47M	76.5	87.9	80.8	76.9	72.7	59.7	69.4	61.8	73.2
RoBERTa-ZINC [40]	480M	79.0	87.0	79.1	79.3	72.5	60.4	71.9	60.6	73.7
Uni-Mol [29]	47M	74.5	84.3	54.8	72.4	68.7	60.5	68.8	60.5	68.1
Enc	47M	**80.2**	88.5	79.1	**79.9**	72.8	60.0	71.4	63.8	74.5
Proj	49M	78.1	88.5	81.5	76.4	69.5	60.2	70.3	61.6	73.3
ProbRead	53M	79.6	**90.2**	89.3	77.9	**82.6**	**62.8**	**77.0**	**68.1**	**78.4**

Table 3: AUC (↑) on six TDC classification datasets (TRIDENT).

	Ames	Carc.	CYP	DILI	hERG	Skin	Avg
BioT5+	75.5	82.9	76.5	83.5	58.3	70.1	74.5
ChemBERTa	78.8	84.5	84.8	**86.1**	70.6	71.6	79.4
SELFormer	79.4	81.1	85.0	80.6	73.2	69.1	78.1
MoLFormer	81.0	86.7	85.5	78.1	71.9	71.7	79.2
RoBERTa	79.2	81.7	85.3	83.5	74.3	63.9	78.0
Uni-Mol	79.1	89.5	83.0	78.6	76.4	62.7	78.2
Enc	79.8	**97.5**	84.8	77.2	75.6	74.3	81.5
Proj	76.6	87.5	82.7	79.5	75.4	71.4	78.8
ProbRead	**84.0**	94.4	**88.1**	85.9	**76.9**	**76.4**	**84.3**

Table 4: RMSE (↓) on four molecular regression datasets (TRIDENT).

	CaCO2	ESOL	FrSlv	Lipo	Avg
BioT5+	0.707	1.823	4.678	1.091	2.075
ChemBERTa	0.613	1.637	3.383	0.912	1.637
SELFormer	0.796	1.725	3.470	1.036	1.757
MoLFormer	0.678	1.637	3.308	0.933	1.639
RoBERTa	0.727	1.647	2.781	0.956	1.528
Uni-Mol	0.754	1.638	3.308	0.983	1.670
Enc	1.040	1.657	4.176	1.041	1.978
Proj	0.870	1.660	3.968	1.030	1.882
ProbRead	**0.502**	**1.355**	**2.198**	**0.792**	**1.212**

These results reveal both the promise and the limitation of static combination. Concatenation cannot regulate the contribution of each information source: the weaker source introduces noise when dimensionalities are imbalanced. More fundamentally, both $\bar{\mathbf{h}}$ and $\mathbf{z}$ are derived from pooled representations. The token-level information lost through fixed pooling is absent from both, and concatenation cannot recover what neither source contains. An effective readout must therefore aggregate directly from $\mathbf{H}$ and adaptively balance the two sources.

4.3 Main Results

Molecular property prediction (TRIDENT). Tables 2 and 3 report AUC on eight MoleculeNet and six TDC classification datasets, and Table 4 reports RMSE on four regression datasets. ProbRead achieves the best average performance in all three tables and the best result on 13 of the 18 datasets. It also outperforms linear probing of larger pretrained models such as RoBERTa-ZINC (480M) and BioT5+ (252M), while using only 53M parameters in total. Compared with the stronger of Enc and Proj, ProbRead improves average AUC from 74.5 to 78.4 on MoleculeNet and from 81.5 to 84.3 on TDC, and reduces average RMSE from 1.882 to 1.212. The largest gains appear on MUV (72.8 to 82.6 AUC) and FreeSolv (3.968 to 2.198 RMSE). At the dataset level, ProbRead outperforms the better of Enc and Proj on 15 of the 18 datasets. These results verify that combining the two representations through ProbRead is more effective than using either one alone or selecting the better one for each dataset.

Protein function prediction (S-PLM). Table 5 reports Fmax on EC and GO for ProbRead, Enc, Proj, and five external protein language models. ProbRead achieves the best Fmax on both tasks. It improves over Enc from 0.750 to 0.856 on EC and from 0.468 to 0.599 on GO, and it exceeds the strongest external baselines, SaProt on EC (0.820) and ProTrek on GO (0.580). Unlike on TRIDENT, the two representations show no reversal here: Proj falls 48% below Enc on EC and 38% below on GO, consistent with the severe erank compression of the S-PLM projector (Section 4.2). The gain of ProbRead therefore does not rely on a strong projected embedding. Moreover, concatenating the two pooled representations barely improves over Enc (Table 1). This suggests that the gain comes from information that the pooled representations do not provide, consistent with our claim that reading encoder tokens directly recovers task-relevant information lost by fixed pooling.

Protein interaction and structural classification (ProteinCLIP). Figure 5 reports AUC on PPI and accuracy on CATH. ProbRead achieves the best result on both tasks (68.36 AUC and 73.64 accuracy) and outperforms all external models, including ESM-2 650M, which is about four times larger than the ProteinCLIP encoder. Notably, the better representation changes between the two tasks: Proj leads on PPI (66.72 vs. 61.95), whereas Enc leads on CATH (68.81 vs. 66.49). ProbRead

Table 5: Fmax (↑) on EC and GO (S-PLM).

Method	EC	GO
ESM-C (300M)	0.787	0.515
ESM-C (600M)	0.789	0.523
ProtST	0.814	0.557
SaProt	0.820	0.576
ProTrek	0.793	0.580
Enc	0.750	0.468
Proj	0.387	0.291
ProbRead	**0.856**	**0.599**

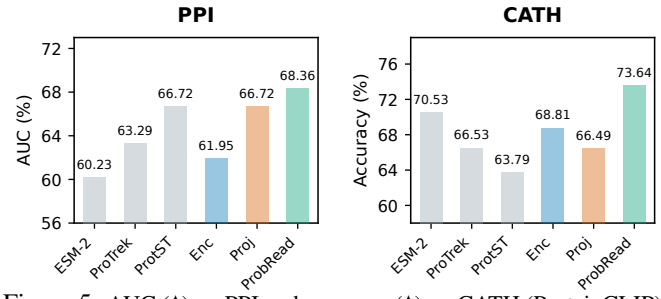

Figure 5: AUC (↑) on PPI and accuracy (↑) on CATH (ProteinCLIP).

exceeds the better of the two on both tasks, supporting our design of adaptively combining both representations rather than fixing either one.

Cross-architecture validation (MoleculeSTM). The three platforms above all use Transformer encoders. To test whether ProbRead also applies to graph encoders, we evaluate it on MoleculeSTM [2], which aligns molecular graphs with text using a five-layer GIN encoder [28]. Table 6 reports AUC on seven MoleculeNet classification datasets. As can be seen, ProbRead outperforms both Enc and Proj on all seven datasets and improves the average AUC from 63.9 (Proj) to 71.3. These results suggest that the benefit of ProbRead is not limited to Transformer encoders.

Table 6: AUC (↑) on seven MoleculeNet classification datasets (MoleculeSTM).

	BACE	BBBP	ClinTox	HIV	MUV	SIDER	Tox21	Avg
Enc	67.1	62.5	48.7	72.3	66.9	58.7	68.7	63.6
Proj	67.0	62.0	53.5	69.9	68.2	58.5	68.3	63.9
ProbRead	**76.2**	**67.1**	**75.4**	**73.9**	**69.9**	**62.2**	**74.3**	**71.3**

4.4 Analysis of Effective Dimensionality

Section 4.2 shows that projection substantially reduces the effective rank of the representation.We examine whether the output of ProbRead recovers the effective rank lost by projection. We compute the erank of the ProbRead output, the pooled encoder embedding, and the projected embedding on six test sets from the three platforms. As we can see from Figure 6, the projected embeddings retain only 7–54% of the encoder's erank, whereas ProbRead recovers 81–123%: 97–102% on TRIDENT, 96% and 123% on S-PLM EC and GO, and 81% on ProteinCLIP PPI. These results show that ProbRead uses the projected embedding without inheriting its low effective rank, consistent with Allosteric Readout reading directly from the encoder tokens.

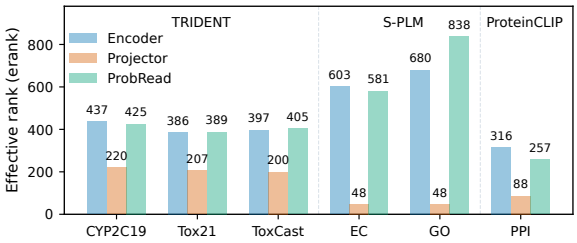

Figure 6: Effective rank of the embeddings extracted by encoder, projector, and ProbRead.

5 Conclusion

We proposed ProbRead, a lightweight readout that jointly uses encoder token embeddings and projected embeddings in frozen cross-modal biomolecular models. Allosteric Readout replaces fixed pooling with token selection guided by the projected embedding, and Homeostatic Fusion adaptively balances the aggregated token representation with the projected embedding. Across three platforms and five groups of biomolecular tasks, ProbRead achieved the best average performance, outperforming encoder-only and projector-only probing as well as linear probing of other biomolecular foundation models. ProbRead also recovered the effective rank lost by projection, reaching 81–123% of the encoder's erank. These results verify that, for frozen cross-modal models, adaptively combining the two representations is more effective than fixing either one in advance.

AI Assistance Disclosure

This paper was written with the assistance of AI language models for polishing English prose, improving clarity, and formatting LaTeX. All scientific ideas, experimental design, data analysis, and interpretations are solely the work of the authors. The AI tool was not used to generate research ideas, conduct experiments, or fabricate results.

Ethics Statement

This work studies how to read out representations from frozen cross-modal contrastive models for molecular and protein property prediction. It involves no human subjects or personally identifiable information, and all experiments use publicly available pretrained models and datasets. ProbRead is a readout for predictive models and does not generate molecules, so we consider its dual-use risk comparable to that of existing property predictors trained on the same public benchmarks. Its predictions are intended for research and have not been validated for clinical or regulatory decisions.

Reproducibility Statement

Sections 3.2.1 and 3.2.2 give the full formulation of ProbRead, and Algorithm 1 in Appendix B summarizes its inference procedure. Section 4.1 describes the platforms, tasks, and training protocol. Appendix C reports the per-platform hyperparameters (Table 8), random seeds, embedding extraction, data splits, and protocols of the linear-probing baselines, and Appendix H defines all evaluation metrics. All pretrained models and datasets used in this work are publicly available. Upon acceptance, we will release our code, including the embedding extraction and evaluation scripts.

References

[1] Feng Jiang, Mangal Prakash, Hehuan Ma, Jianyuan Deng, Yuzhi Guo, Amina Mollaysa, Tommaso Mansi, Rui Liao, and Junzhou Huang. TRIDENT: Tri-modal molecular representation learning with taxonomic annotations and local correspondence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[2] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. Multi-modal molecule structure-text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.

[3] Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuning Jiang, Qiang Shao, Jiangshan Chen, and Dong Xu. S-PLM: Structure-aware protein language model via contrastive learning between sequence and structure. *Advanced Science*, 12(5):e2404212, 2025.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

[5] Le Zhang, Qian Yang, and Aishwarya Agrawal. Assessing and learning alignment of unimodal vision and language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14604–14614, 2025.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

[7] Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. ProtST: Multi-modality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning (ICML)*, 2023.

[8] Kevin Wu. ProteinCLIP: Multi-modal learning for aligning protein sequence, structure, and function. 2024. GitHub: https://github.com/wukevin/proteinclip.

[9] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[10] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. In *Nature Machine Intelligence*, volume 4, pp. 1256–1264, 2022.

[11] Paul Pu Liang, Zihao Deng, Martin Q. Ma, James Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized contrastive learning: Going beyond multi-view redundancy. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[12] Zhuo Ouyang, Kaiwen Hu, Qi Zhang, Yifei Wang, and Yisen Wang. Projection head is secretly an information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2025.

[13] Florian Bordes, Randall Balestriero, and Yann LeCun. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research*, 2023.

[14] Yihao Xue, Eric Gan, Jiayi Ni, Siddharth Joshi, and Baharan Mirzasoleiman. Investigating the benefits of projection head for representation learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

[17] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations (ICLR)*, 2022.

[18] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3D infomax improves GNNs for molecular property prediction. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[19] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4:279–287, 2022.

[20] Jin Su, Chenchen Han, Yuyang Zhou, et al. ProTrek: Navigating the protein universe through tri-modal contrastive learning. *Nature Biotechnology*, 2025.

[21] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations (ICLR)*, 2023.

[22] Bowen Gao, Bo Qiang, Haichuan Tan, Yinjun Jia, Minsi Ren, Minsi Lu, Jingjing Liu, Wei-Ying Ma, and Yanyan Lan. DrugCLIP: Contrastive protein-molecule representation learning for virtual screening. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[23] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

[24] Atakan Yüksel, Erva Ulusoy, Berk Erman, and Ayça Begüm Tosun. SELFormer: Molecular representation learning via SELFIES language models. *Machine Learning: Science and Technology*, 4(2), 2023.

[25] Jannis Born, Matteo Manica, et al. SMI-TED: Large-scale pre-training of a foundation model for materials and chemistry. *Communications Chemistry*, 2025.

[26] Feng Luo et al. ChemFM: A foundation model for chemical design and property prediction. *Communications Chemistry*, 2024.

[27] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1263–1272, 2017.

[28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[29] Xiaohong Zhou et al. Uni-Mol2: Exploring molecular pretraining model at scale. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[30] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[31] EvolutionaryScale. Esm cambrian: Protein language modeling with metagenomics. In *bioRxiv preprint*, 2024.

[32] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehber, 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.

[33] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell Systems*, 12(6):654–669, 2021.

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

[35] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

[36] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

[37] Vladimir Gligorijevic, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1): 3168, 2021.

[38] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P. Waman, Paul Ashford, Harry M. Scholes, Camilla S. M. Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. CATH: increased structural coverage of functional space. *Nucleic Acids Research*, 49(D1):D266–D273, 2021.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] Qizhi Pei, Lijun Zhang, Liang Wu, et al. BioT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning. *Findings of the Association for Computational Linguistics (ACL)*, 2024.

[42] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, pp. 606–610, 2007.

[43] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[44] Yu Gui, Cong Ma, and Yiqiao Zhong. Unraveling projection heads in contrastive learning: Insights from expansion and shrinkage. *arXiv preprint arXiv:2306.03335*, 2023.

[45] Jiajun Ma, Tianyang Hu, and Wenjia Wang. Deciphering the projection head: Representation evaluation self-supervised learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4724–4732, 2024.

[46] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 9929–9939, 2020.

[47] Benoit Dufumier, Javiera Castillo-Navarro, Devis Tuia, and Jean-Philippe Thiran. What to align in multimodal contrastive learning? In *International Conference on Learning Representations (ICLR)*, 2025.

[48] Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

[49] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[50] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2020.

[51] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[52] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9578–9588, 2021.

[53] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[54] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 12310–12320, 2021.

[55] Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[56] Yifan Zhang, Zhiquan Tan, Jingqin Yang, Weiran Huang, and Yang Yuan. Matrix information theory for self-supervised learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[57] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant input. In *International Conference on Machine Learning*, 2019.

[58] Andrew Jaegle, Felix Gimeno, Andrew Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

[59] Chankyu Lee, Rajarshi Roy, Menber Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Han. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *International Conference on Learning Representations (ICLR)*, 2025.

A Additional Related Work

Projection heads and information loss. The impact of projection heads on representation quality has been widely studied. Chen et al. [6] first documented the phenomenon; Bordes et al. [13] showed that removing projection layers at evaluation can substantially improve LP accuracy; and Ouyang et al. [12] formalized it via information bottleneck theory. These studies primarily examine projection removal and evaluation strategies. A second line of work provides theoretical explanations. Xue et al. [14] showed that deeper layers progressively specialize representations, with lower layers retaining richer feature diversity. Gui et al. [44] identified expansion–shrinkage effects, and Ma et al. [45] linked the projector to the uniformity objective [46]. On the multimodal side, Liang et al. [11] showed that contrastive learning captures only shared information (S), discarding modality-unique features ($U_1 + U_2$), and Dufumier et al. [47] further analyzed multimodal interactions via partial information decomposition [48], identifying shared, unique, and synergistic components. This connects to the InfoMin principle [49] and multi-view information bottleneck theory [50]. Geometrically, the projection bottleneck appears as dimensional collapse [51; 52], where representations cluster onto low-dimensional submanifolds; pretraining remedies [53; 54; 55] mitigate this but require modifying training. Zhang et al. [56] established $\exp(S) =$ erank via matrix information theory, connecting effective rank to the exponential of von Neumann entropy. Existing studies mainly characterize projection-related information loss or modify pretraining procedures. Post-hoc regulation of complementary information sources in frozen models remains underexplored.

Token aggregation. Transformer encoders produce variable-length token sequences, yet downstream tasks require a fixed-size representation. The standard approach, using CLS-token selection or mean pooling, applies a task-agnostic aggregation that may discard positional and local structure. Learned alternatives address this limitation: set attention with inducing points [57], cross-attention with learnable latent arrays [58], and latent attention pooling [59]. CLIP's visual encoder likewise replaces mean pooling with attention-based pooling [4]. However, these methods study token aggregation as an independent architectural choice and do not consider the interaction between pooling and projection in frozen contrastive models, particularly the role of the projector's alignment signal in regulating token-level information selection.

B Algorithmic

Algorithm 1 summarizes the inference procedure of ProbRead: the frozen model provides the token embeddings and the projected embedding, Allosteric Readout aggregates the tokens using the projected embedding as the query, and Homeostatic Fusion combines the aggregated representation with the projected embedding for prediction.

C Implementation Details

C.1 Models and Datasets

We evaluate on three contrastive platforms spanning molecular and protein domains. TRIDENT [1] aligns molecular SMILES, text descriptions, and hierarchical taxonomic annotations. Its molecular branch uses the 47M-parameter MoLFormer-XL [10], a Transformer with linear attention, to produce 768-dimensional representations, which a three-layer MLP maps into a 512-dimensional shared space. We evaluate it on 14 classification and 4 regression datasets from MoleculeNet and TDC. S-PLM [3] aligns protein sequences with structural representations derived from contact maps.

Algorithm 1 ProbRead inference.

Require: Input $\mathbf{x}_m$; frozen encoder $f_{\boldsymbol{\theta}_m}$; pooling operator P_m; frozen projector $g_{\boldsymbol{\phi}_m}$;
Require: Trained parameters $\{\mathbf{W}_Q^{(a)}, \mathbf{W}_K^{(a)}, \mathbf{W}_V^{(a)}\}_{a=1}^{n_h}$, $\mathbf{W}_O$, $\mathbf{W}_p$, $\mathbf{W}_g$, and prediction head $c_{\boldsymbol{\omega}}$;
Ensure: Prediction $\hat{y}$;

$\mathbf{H}_m \leftarrow f_{\boldsymbol{\theta}_m}(\mathbf{x}_m) = [\mathbf{h}_{m,1}, \ldots, \mathbf{h}_{m,L_m}]$ ▷ Token representations
$\bar{\mathbf{h}}_m \leftarrow P_m(\mathbf{H}_m)$ ▷ Fixed pooled representation
$\mathbf{z}_m \leftarrow g_{\boldsymbol{\phi}_m}(\bar{\mathbf{h}}_m)$ ▷ Projector representation

// allosteric readout
for $a = 1, \ldots, n_h$ **do**
  $\mathbf{q}_m^{(a)} \leftarrow \mathbf{W}_Q^{(a)} \mathbf{z}_m$ ▷ Regulatory query
  for $i = 1, \ldots, L_m$ **do**
    $\mathbf{k}_{m,i}^{(a)} \leftarrow \mathbf{W}_K^{(a)} \mathbf{h}_{m,i}, \quad \mathbf{v}_{m,i}^{(a)} \leftarrow \mathbf{W}_V^{(a)} \mathbf{h}_{m,i}$
    $e_{m,i}^{(a)} \leftarrow {\mathbf{q}_m^{(a)}}^{\top} \mathbf{k}_{m,i}^{(a)} / \sqrt{d_h}$
  end for
  $\alpha_{m,i}^{(a)} \leftarrow \dfrac{\exp(e_{m,i}^{(a)})}{\sum_{j=1}^{L_m} \exp(e_{m,j}^{(a)})} \quad \text{for } i = 1, \ldots, L_m$
  $\mathbf{o}_m^{(a)} \leftarrow \sum_{i=1}^{L_m} \alpha_{m,i}^{(a)} \mathbf{v}_{m,i}^{(a)}$ ▷ Regulated token retrieval
end for
$\hat{\mathbf{h}}_{\mathrm{AR},m} \leftarrow \mathrm{LayerNorm}\Big(\mathbf{W}_O[\mathbf{o}_m^{(1)}; \ldots; \mathbf{o}_m^{(n_h)}]\Big)$

// homeostatic fusion
$\mathbf{p}_m \leftarrow \mathbf{W}_p \mathbf{z}_m$ ▷ Transform projector features
$\mathbf{g}_m \leftarrow \sigma\Big(\mathbf{W}_g[\hat{\mathbf{h}}_{\mathrm{AR},m}; \mathbf{z}_m]\Big)$ ▷ Input-conditioned balance
$\hat{\mathbf{h}}_{\mathrm{out},m} \leftarrow \mathbf{g}_m \odot \hat{\mathbf{h}}_{\mathrm{AR},m} + (\mathbf{1} - \mathbf{g}_m) \odot \mathbf{p}_m$ ▷ Complementary feature fusion
$\hat{y} \leftarrow c_{\boldsymbol{\omega}}(\hat{\mathbf{h}}_{\mathrm{out},m})$ ▷ Task prediction
return $\hat{y}$

Its sequence branch augments the 33-layer ESM2-t33-650M-UR50D model [30] with Houlsby-style structure-aware adapters inserted after the self-attention and feed-forward blocks of the upper Transformer layers. Mean-pooled 1280-dimensional representations are mapped by a two-layer MLP into a 256-dimensional shared space. We evaluate it on EC and GO function prediction. ProteinCLIP [8] aligns protein sequences with functional text descriptions. Its protein branch uses the 30-layer ESM-2 150M model [30] to produce 640-dimensional residue representations, which are mean-pooled and linearly projected into a normalized 128-dimensional shared space. We evaluate it on PPI interaction prediction and CATH structural classification. All experiments use linear probing with frozen encoders and projectors, averaged over three seeds.

The encoder f_θ and projector g_ϕ remain *frozen* throughout. Only the query projection W_q, readout parameters, gating parameters (W_p, W_g), LayerNorm, and the MLP head are trained with AdamW (lr=10^{-4}, weight decay=10^{-4}) and cosine annealing with early stopping. The trainable parameters account for fewer than 10% of the encoder parameters across the three evaluated platforms (0.8–8.7%). Complete training configurations, embedding extraction details, and LP baseline protocols for all platforms are provided in Appendix C to facilitate reproducibility.

C.2 Platform Descriptions

Table 7 summarizes the three evaluation platforms. All follow the shared projection space paradigm: modality-specific encoders are pretrained with contrastive losses, and the projection head is discarded at evaluation. Encoder and projector dimensions are specified in Table 8.

Table 7: Evaluation platform overview.

Platform	Encoder	Params	Modalities	Downstream tasks
TRIDENT	MoLFormer-XL	47M	SMILES + Text + HTA	14 cls + 4 reg (MoleculeNet, TDC)
S-PLM	ESM-2 + Adapter-H	650M	Protein seq. + 3D struct.	EC, GO function prediction
ProteinCLIP	ESM-2 t30	150M	Protein + Text	PPI interaction, CATH classification

TRIDENT [1] aligns molecular SMILES, text descriptions, and hierarchical taxonomic annotations (HTA) via a GRAM volume-based contrastive loss. The MoLFormer-XL encoder was pretrained on 1.1B SMILES strings with linear attention. We evaluate on 14 classification (AUC) and 4 regression (RMSE) datasets from MoleculeNet and TDC benchmarks.

S-PLM [3] aligns protein sequences with 3D structures using a contrastive objective. The encoder consists of a frozen ESM-2 (650M) backbone augmented with trainable Adapter-H side-networks (52.6M adapter parameters). We evaluate on EC (enzyme classification, 538 classes) and GO (gene ontology molecular function, 489 classes).

ProteinCLIP [8] aligns proteins with functional text descriptions using a CLIP-style contrastive loss, with ESM-2 t30 (150M, 30 layers) as the sequence encoder. We evaluate on PPI binary classification and CATH architecture-level structural classification (43 classes).

C.3 Training Configuration

ProbRead adds a lightweight readout module on top of frozen encoder and projector weights. Table 8 provides the complete training configuration for each platform, sufficient to reproduce our results given the frozen encoder/projector checkpoints and the ProbRead module code.

Table 8: ProbRead training configuration. All encoders and projectors remain fully frozen throughout. ProteinCLIP runs all epochs and tracks the best validation checkpoint; other platforms use early stopping with the specified patience. The MLP head is a single linear layer. d_e and d_p together with n_{heads} fully determine the ProbRead module architecture.

	TRIDENT	S-PLM	ProteinCLIP
d_e (encoder dim)	768	1280	640
d_p (projector dim)	512	256	128
n_{heads}	8	8	8
Batch size	32	16	128 (pairs)
Epochs	100	100	50
Early stopping	15	15	—
Learning rate	10^{-4}	10^{-4}	10^{-4}
Weight decay	10^{-4}	10^{-4}	10^{-4}
Scheduler	Cosine annealing ($T_{\max}$ = epochs)		
Optimizer	AdamW		
Seeds	3 (0, 1, 2)		
MLP head	Linear(d_e, n_{classes}) or Linear(d_e, 1) for regression		
Trainable params	4.1M	4.9M	1.6M
% of encoder	8.7%	0.8%	1.1%

Embedding extraction. For all platforms, we pre-extract and cache encoder token sequences $\mathbf{H}$, pooled representations $\bar{\mathbf{h}}$, and projector outputs $\mathbf{z}$ from the frozen models before training. This avoids redundant forward passes through the encoder during ProbRead training. TRIDENT uses CLS token pooling; S-PLM and ProteinCLIP use mean pooling over non-padding tokens.

LP baselines. For the Enc and Proj LP baselines on TRIDENT, we use scikit-learn's LogisticRegression (classification, C=1.0, LBFGS solver, max_iter=1000) and Ridge regression (α=1.0) with standard scaling, consistent with the original TRIDENT evaluation protocol. For S-PLM and ProteinCLIP, we train a single-layer neural LP head with the same AdamW optimizer and cosine scheduler as ProbRead, ensuring a fair comparison.

Data splits. TRIDENT MoleculeNet datasets use the original scaffold splits provided by MoleculeNet; TDC datasets use scaffold splits generated by the TDC library with 3 seeds. S-PLM uses the official EC/GO train/valid/test splits. ProteinCLIP PPI uses a random 80/10/10 split; CATH uses the topology-level split ensuring no structural leakage.

D Effective Rank Data

Tables 9–10 provide the exact numerical data underlying Figures 4 and 6 in the main text. All datasets satisfy $N > d_e > d_p$, ensuring the erank upper bound is governed by dimensionality rather than sample size.

Table 9: Encoder vs. projector effective rank (data for Figure 4).

Platform	Dataset	N	Enc. erank	Proj. erank
TRIDENT	CYP2C19	1,267	436.7	220.2
TRIDENT	Tox21	784	386.3	206.7
TRIDENT	ToxCast	858	397.3	200.3
S-PLM	EC	1,919	602.7	48.1
S-PLM	GO	3,416	680.1	47.8
ProteinCLIP	PPI	11,018	315.9	87.5

Table 10: ProbRead effective rank analysis (data for Figure 6). Recovery % = ProbRead erank / Enc erank. On S-PLM GO, ProbRead *exceeds* the encoder, indicating regulated readout utilizes dimensions more effectively than fixed mean pooling.

Platform	Dataset	Enc.	Proj.	ProbRead	Recovery %
TRIDENT	CYP2C19	436.7	220.2	425.2	97%
TRIDENT	Tox21	386.3	206.7	389.0	101%
TRIDENT	ToxCast	397.3	200.3	404.6	102%
S-PLM	EC	602.7	48.1	581.0	96%
S-PLM	GO	680.1	47.8	837.5	123%
ProteinCLIP	PPI	315.9	87.5	257.4	81%

E TRIDENT Per-Dataset Results

E.1 Concatenation Diagnostic

The main text (Table 1) reports only average performance for the concatenation diagnostic. Table 11 provides per-dataset classification results; Table 12 provides regression results. Concat improves on 7/14 classification datasets and 2/4 regression datasets over the stronger of Enc and Proj, indicating widespread but inconsistent complementarity.

Table 11: TRIDENT per-dataset classification (AUC↑). Concat improves on 7/14 datasets over max(Enc, Proj). The largest Concat gains appear on tasks where Enc and Proj carry distinct strengths (e.g., Tox21: +2.5 over Enc, DILI: +1.4 over Proj). Failures occur when one source strongly dominates (e.g., Carc: Enc 97.5 ≫ Proj 87.5, Concat drops to 93.4 due to noise from the weaker source).

	Ames	BACE	BBBP	Carc.	ClinTox	CYP	DILI	hERG	HIV	MUV	SIDER	Skin	Tox21	ToxCast	Avg
Enc	79.8	**80.2**	88.5	**97.5**	79.1	**84.8**	77.2	**75.6**	**79.9**	**72.8**	60.0	**74.3**	71.4	63.8	**77.49**
Proj	76.6	78.1	88.5	87.5	81.5	82.7	79.5	75.4	76.4	69.5	60.2	71.4	70.3	61.6	75.65
Concat	**80.3**	80.1	**89.5**	93.4	**81.8**	**84.8**	**80.9**	74.4	77.7	70.9	**61.0**	69.3	**73.9**	**64.1**	77.31

E.2 S-PLM Numerical Data

Table 13 provides the full numerical data for Table 5 in the main text.

Table 12: TRIDENT per-dataset regression (RMSE↓). Concat achieves the best average but not on all datasets: Lipo regression benefits more from the projector alone, whereas FreeSolv sees the largest Concat improvement (-0.756 vs Proj).

	CaCO2	ESOL	FreeSolv	Lipo	Avg
Enc	1.040	1.657	4.176	1.041	1.978
Proj	**0.870**	1.660	3.968	**1.030**	1.882
Concat	0.984	**1.589**	**3.212**	1.093	**1.720**

Table 13: S-PLM function prediction (LP, Fmax↑). Numerical data for Table 5.

Method	EC (Fmax↑)	GO (Fmax↑)
ESM-C 300M	0.787	0.515
ESM-C 600M	0.789	0.523
ProtST	0.814	0.557
SaProt 650M	0.820	0.576
ProTrek 650M	0.793	0.580
Encoder representation ($d = 1280$)	0.750	0.468
Projector representation ($d = 256$)	0.387	0.291
ProbRead	**0.856**	**0.599**

F MoleculeSTM: Cross-Architecture Validation

To verify that ProbRead generalizes beyond Transformer encoders, we evaluate on MoleculeSTM, which uses a five-layer GIN graph neural network with approximately 0.5 million parameters to produce 300-dimensional molecular representations, using an architecture fundamentally different from the Transformers in the main text. MoleculeSTM aligns molecular graphs with text descriptions through contrastive learning, using a projection head that maps the 300-dimensional GIN representations into a 256-dimensional shared space.

ProbRead improves average AUC from 63.6 to 71.3 (+7.7), winning on all 7 datasets. The largest gain is on ClinTox (+26.7 over Enc), where ProbRead effectively balances the projector's complementary toxicity-relevant features with encoder token information.

G External Baseline Detailed Results

The main text shows TRIDENT results in per-dataset tables (Tables 2–4) and S-PLM results in Table 5. Table 14 below provides the ProteinCLIP per-task baseline comparison, supplementing Figure 5 with exact numerical values.

Table 14: Protein baselines across S-PLM and ProteinCLIP tasks. ProbRead surpasses all external baselines on all four tasks. On CATH, ProbRead exceeds the four-times-larger ESM-2 650M by +3.1pp.

Method	S-PLM		ProteinCLIP	
	EC (Fmax↑)	GO (Fmax↑)	PPI (AUC↑)	CATH (Acc↑)
ESM-C 300M	0.787	0.515	—	—
ESM-C 600M	0.789	0.523	—	—
ProtST	0.814	0.557	66.72	63.79
SaProt 650M	0.820	0.576	—	—
ProTrek 650M	0.793	0.580	63.29	66.53
ESM-2 650M	—	—	60.23	70.53
Enc	0.750	0.468	61.95	68.81
Proj	0.387	0.291	66.72	66.49
ProbRead	**0.856**	**0.599**	**68.36**	**73.64**

H Evaluation Metrics

We use standard metrics for each task type:

AUC (Area Under the ROC Curve). For binary and multi-label classification tasks (TRIDENT, ProteinCLIP PPI). AUC measures the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example. Multi-label datasets (e.g., Tox21 with 12 tasks) report the average AUC across all tasks. Values range from 0 to 100 (reported as percentages), with 50 corresponding to random performance.

RMSE (Root Mean Squared Error). For regression tasks (TRIDENT: CaCO2, ESOL, FreeSolv, Lipophilicity). RMSE $= \sqrt{\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2}$. Lower is better.

Fmax (Maximum F1 Score). For multi-label protein function prediction (S-PLM: EC, GO). Fmax is the maximum protein-centric F1 score over all possible thresholds: $F_{\text{max}} = \max_\tau x_\tau \frac{2 \cdot \text{precision}(\tau) \cdot \text{recall}(\tau)}{\text{precision}(\tau) + \text{recall}(\tau)}$, where precision and recall are averaged across all proteins at threshold τ. This is the standard metric used in CAFA evaluations.

Accuracy. For multi-class classification (ProteinCLIP CATH, 43 architecture-level classes). Standard top-1 classification accuracy.

I Limitations and Future Work

Like other methods built on pretrained deep models, ProbRead can only use information that the frozen encoder and projector already capture. If pretraining misses task-relevant features, a better readout cannot recover them. Future work includes optimizing the readout jointly with contrastive pretraining.