REVERSE DISTILLATION: DISENTANGLING AND SCALING PROTEIN LANGUAGE MODEL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlike the foundation model scaling laws seen in natural language processing and computer vision, biological foundation models scale relatively poorly. For example, the ESM-2 family of protein language models plateaus at 650M-3B parameters on ProteinGym benchmarks. We address this limitation by introducing *Reverse Distillation*, a principled framework that decomposes large protein language model representations into orthogonal subspaces guided by smaller models of the same family. We hypothesize that this decomposition matches the natural hierarchy of protein properties, where broad features like secondary structure are robustly captured by compact, smaller models while the residual capacity of larger models specializes in protein-family specific functions. Our method is theoretically grounded and enables monotonic scaling—larger reverse-distilled models consistently outperform their smaller counterparts, overcoming the scaling plateau. Moreover, on ProteinGym benchmarks, reverse-distilled ESM-2 variants broadly outperform their respective baseline models at the same embedding dimensionality. Our approach offers a generalizable framework for disentangling hierarchical feature spaces in foundation model embeddings, with potential applications across biology and other domains where scaling challenges persist.

1 INTRODUCTION

Protein language models (PLMs) have emerged as powerful representation learners, capturing evolutionary patterns from millions of sequences and enabling unprecedented capabilities in structure prediction (Lin et al., 2023), function annotation (Yu et al., 2023), and protein design (Ferruz et al., 2022; Devkota et al., 2024). These models learn rich protein representations through self-supervised training on vast sequence databases. However, unlike the predictable scaling laws observed in natural language processing (Kaplan et al., 2020; Hoffmann et al., 2022), PLMs—and more broadly, biological foundation models—exhibit counterintuitive scaling behavior: larger models often underperform smaller ones on functional prediction tasks (Li et al., 2024). For example, on ProteinGym deep mutational scanning (DMS) benchmarks, the ESM-2 family peaks at 650M-3B parameters, with the 15B model showing degraded performance.

This unexpected scaling behavior creates fundamental challenges. Given models M_1, M_2 with $|M_2| > |M_1|$ parameters, we observe non-monotonic performance: we often find that smaller models outperform larger models on downstream tasks. Moreover, we cannot reliably predict which biological tasks will exhibit poor scaling behavior, leading to difficulties in model selection for any specific task. A related limitation of PLMs is that embeddings across model scales are unrelated. In contrast, Matryoshka-style embeddings (Kusupati et al., 2024) in natural language processing are structured such that their prefixes are also directly usable. With these embeddings, prefixes of an overall embedding are themselves functional, albeit with some performance degradation. This enhances computational and storage efficiency, enabling an "embed once, reuse prefixes as needed" paradigm. However, current PLM representations do not offer this advantage: representations of dimension k cannot be truncated to dimension $k' < k$ while maintaining smooth performance degradation.

We interpret these scaling behaviors through a bias-variance lens. We hypothesize that small PLMs, constrained by capacity, preferentially encode frequent, low-complexity biological regularities, e.g., secondary structure propensities, hydrophobicity etc. As capacity grows, models can represent rarer, higher-order phenomena: family-specific patterns, epistatic interactions, allosteric signals, dynamics etc. However, when these specialized signals are entangled with universal features[1] in a single representational space, the entanglement increases variance in downstream linear evaluations that form the basis of most task-specific predictors. This entanglement may explain why naive scaling fails: task-irrelevant specialized features add noise to the universal patterns that drive performance on most benchmarks.

We introduce **Reverse Distillation**[2], a principled framework that systematically decomposes large PLM representations into interpretable, orthogonal subspaces anchored by smaller models. Unlike traditional knowledge distillation that compresses large models into small ones, our method identifies what unique information each model scale contributes. The key insight is structural: by treating smaller model representations as a basis and extracting orthogonal residuals, we prevent destructive interference between universal and specialized features.

Formally, given models M_r and M_p where $|M_r| < |M_p|$, with embedding dimensionality $k_r < k_p$, we decompose representations as $H_p \approx [H_r, H_{res}]$ where $H_r \in \mathbb{R}^{n \times k_r}$ captures universal patterns and $H_{res} \in \mathbb{R}^{n \times (k_p - k_r)}$ captures specialized information orthogonal to H_r. We also prove this decomposition is MSE-optimal among all k_p-dim representations that fully encompass M_r (i.e., the cylinder set of M_r in $\mathbb{R}^{k_p}$).

Our contributions are:

- **Hierarchical Decomposition**: We show how to transform a family of PLM models such that they follow a hierarchical structure where each higher scale adds orthogonal information. Our decomposition is also guaranteed to approximate the original representation space well.
- **Matryoshka-style Embeddings and Monotonic Improvement**: Reverse-distilled embeddings are constructed such that nested embeddings of dimensionality d contain prefixes of sizes $d_1 < d_2 < d_3 < d$ such that each prefix is the reverse-distilled embedding of the corresponding dimensionality. Our decomposition thus provides controlled performance degradation as a function of embedding size.
- **Scaling Consistency**: Reverse distillation scales nearly always, i.e., larger reverse-distilled models consistently perform better than smaller ones.
- **Improvement over Baseline**: For the ESM-2 family, reverse-distilled models of the same embedding size (e.g., 1280 for ESM-2 650M) generally outperform their corresponding baselines, particularly for tasks requiring specialized features.

2 Method

Motivation and Intuition The ESM-2 family spans embedding dimensions from 320 (8M parameters) to 5120 (15B parameters), providing a systematic testbed for analyzing PLM scaling behavior. Each model learns residue co-evolution patterns through self-attention mechanisms, but capacity constraints induce different feature distributions across model scales. Small models operate under severe capacity constraints. To minimize training perplexity, these models must prioritize the most frequently occurring co-evolutionary patterns—those that maximize compression across the protein sequence distribution. We hypothesize these

[1] We use “universal” and “specialized” as convenient descriptors, though these are necessarily approximations. The boundary between these categories is fluid and context-dependent, but the distinction is useful for discussing scaling behavior.

[2] The term “reverse distillation” has appeared in certain teacher-student architectures in ML literature. Our usage—decomposing large models using smaller ones as a basis—is distinct and we believe intuitive from context.

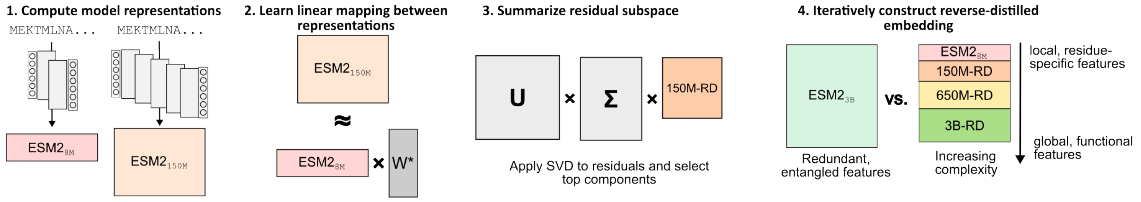
1. Compute model representations
MEKTMLNA...
MEKTMLNA...
ESM2 8M
ESM2 150M
2. Learn linear mapping between representations
ESM2 150M
≈
ESM2 8M
×
W*
3. Summarize residual subspace
U
×
Σ
×
150M-RD
Apply SVD to residuals and select top components
4. Iteratively construct reverse-distilled embedding
ESM2 3B
vs.
ESM2 8M
150M-RD
650M-RD
3B-RD
local, residue-specific features
global, functional features
Redundant, entangled features
Increasing complexity

Figure 1: **Overview of Reverse Distillation** Large protein language models (e.g., ESM2$_{3B}$) entangle universal and specialized features in a single representational space, hindering the performance of downstream linear probes. Reverse distillation constructs a product space by preserving the smaller model's representation (capturing universal features) and extracting orthogonal residuals via SVD (capturing specialized features unique to the larger model). Iterating this process across a model family yields Matryoshka-style embeddings where each prefix corresponds to a valid reverse-distilled representation at that scale.

patterns correspond to universal protein properties: secondary structure propensities, hydrophobicity patterns, and conserved structural motifs. The 8M model lacks sufficient capacity to represent rare, family-specific patterns; its limited parameters are allocated to features with the highest marginal utility across the training distribution.

Large models possess capacity for both universal and specialized patterns. They can encode enzyme-specific catalytic motifs, protein family-specific allosteric couplings, and higher-order interaction patterns. However, as Li et al. (2024) demonstrated, downstream performance often relies on early, low-level features rather than additional capacity; consequently, linear probes on larger, mixed representations frequently fail to isolate task-relevant signal from task-irrelevant variance.

Our key intuition is that smaller models provide a natural basis for disentanglement (Fig. 1). A smaller model from the same family—trained on identical data with the same architecture—produces representations biased toward universal features due to capacity constraints. By computing the orthogonal complement of the smaller model's subspace within the larger model's representation, we achieve partial separation of universal and specialized features without requiring explicit feature identification. We employ linear decomposition throughout our framework to maintain interpretability. Linear methods reveal directly accessible information in representations without confounding from nonlinear prediction heads, and enable precise attribution of information to specific model scales. While nonlinear methods might achieve higher downstream performance, linear decomposition provides the analytical tractability necessary to characterize how biological information distributes across the model hierarchy.

2.1 Problem Formulation

Consider a hierarchy of protein language models $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$ ordered by parameter count. Each model M_i maps sequences to embeddings:

$$M_i : \mathcal{P}^* \to \mathbb{R}^{n \times k_i}$$

where $\mathcal{P}$ is the amino acid alphabet, n varies per sequence, and $k_1 < k_2 < \ldots < k_m$ are embedding dimensions.

For the ESM-2 family, this hierarchy exists naturally: 8M (k_1 = 320), 35M (k_2 = 480), 150M (k_3 = 480), 650M (k_4 = 1280), 3B (k_5 = 2560), 15B (k_6 = 5120).

While our framework does not require monotonically increasing dimensions—appropriate dimensionality reduction via PCA or variational autoencoders could enable reverse distillation between arbitrary model pairs—the ESM-2 family's architecture provides this structure directly, simplifying our implementation.

For a ProteinGym DMS Dataset $\mathcal{D} = \{s_i\}_{i=1}^{N}$ with sequence lengths $\{n_i\}$, the total amino acid positions $L = \sum_{i=1}^{N} n_i$ represents our effective sample size for learning linear relationships. This formulation (treating all positions as samples) enables data-efficient subspace learning. For training, we used $N = 10,000$ sequences sampled randomly from UniRef50; all sequences had 30% or lower sequence identity to datasets used in Section 3. Due to the simple linear transforms involved in this work, this N was sufficient: as an ablation, we computed the difference between the experimental results obtained from the reversed distilled models trained on 10,000 vs 1,000 sequences; they differed by less than 0.01% in Spearman correlation.

Reverse Distillation Decomposition

Definition 1 (Reverse Distillation Decomposition)**.** *Given models M_r and M_p where $r < p$, we decompose the representation space $\mathbb{R}^{n \times k_p}$ into orthogonal subspaces:*

$$\mathcal{S}_p = \mathcal{S}_r \oplus \mathcal{S}_{res}$$

where $\mathcal{S}_r \cong \mathbb{R}^{n \times k_r}$ preserves M_r's representations and $\mathcal{S}_{res} \cong \mathbb{R}^{n \times (k_p - k_r)}$ captures orthogonal residual information.

We express any representation $H_p \in \mathbb{R}^{n \times k_p}$ as:

$$H_p \approx [H_r, H_{res}]$$

where $H_r \in \mathbb{R}^{n \times k_r}$ comes directly from the smaller model M_r, and $H_{res} \in \mathbb{R}^{n \times (k_p - k_r)}$ represents the unique contribution of the larger model.

We preserve entire smaller models rather than selecting subsets of their dimensions. This choice, enabled by the natural progression of embedding sizes (320→640→1280→2560→5120 in ESM-2), maintains interpretability—we know H_r represents the complete "universal" feature space learned by the smaller model, making the residual space $\mathcal{S}_{res}$ directly interpretable as specialized features.

2.2 Algorithms

Algorithm 1 presents our training procedure.

Algorithm 1 Reverse Distillation Algorithm (Pre-training)

```
Require: Dataset $\mathcal{D} = \{s_i\}_{i=1}^N$ where $|s_i| = n_i$, models $M_r$, $M_p$ with $r < p$
Ensure: Subspace decomposition matrices $\mathbf{W}^*$, $\mathbf{V}_{res}$
Phase 1: Compute Representations
for $i = 1$ to $N$ do
    $H_r^{(i)} = M_r(s_i) \in \mathbb{R}^{n_i \times k_r}$ {Variable length $n_i$}
    $H_p^{(i)} = M_p(s_i) \in \mathbb{R}^{n_i \times k_p}$
end for
Phase 2: Learn Linear Mappings
Define total length: $L = \sum_{i=1}^N n_i$
Stack representations: $\tilde{H}_r = \text{vstack}(H_r^{(1)}, \ldots, H_r^{(N)}) \in \mathbb{R}^{L \times k_r}$
Stack representations: $\tilde{H}_p = \text{vstack}(H_p^{(1)}, \ldots, H_p^{(N)}) \in \mathbb{R}^{L \times k_p}$
Solve: $\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\tilde{H}_p - \tilde{H}_r \mathbf{W}\|_F^2$
Compute residuals: $\mathbf{R} = \tilde{H}_p - \tilde{H}_r \mathbf{W}^* \in \mathbb{R}^{L \times k_p}$
Phase 3: Subspace Identification
Apply SVD: $\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
Select top $(k_p - k_r)$ components: $\mathbf{V}_{res} = \mathbf{V}[:, 1 : (k_p - k_r)]$
return $\mathbf{W}^*$, $\mathbf{V}_{res}$
```

Algorithm 2 shows inference. The decomposed representation $H_{rd} = [H_r, H_{res}]$ is Matryoshka by construction—prefixes correspond to valid smaller model outputs, enabling adaptive compute at deployment.

Algorithm 3 extends to entire hierarchies. Chaining reveals hierarchical structure where each scale contributes orthogonal information that cannot be linearly predicted from smaller models.

Algorithm 2 Reverse Distillation Inference

```
Require: New sequence $s$ with $|s| = n$, learned matrices $\mathbf{W}^*$, $\mathbf{V}_{res}$, models $M_r$, $M_p$
Ensure: Decomposed representation $H_{rd} \in \mathbb{R}^{n \times k_p}$
$H_r = M_r(s) \in \mathbb{R}^{n \times k_r}$ {Smaller model embedding}
$H_p = M_p(s) \in \mathbb{R}^{n \times k_p}$ {Larger model embedding}
$H_{pred} = H_r \mathbf{W}^* \in \mathbb{R}^{n \times k_p}$ {Predicted large model embedding}
$R = H_p - H_{pred} \in \mathbb{R}^{n \times k_p}$ {Unexplained residuals}
$H_{res} = R\mathbf{V}_{res} \in \mathbb{R}^{n \times (k_p - k_r)}$ {Projected residuals}
$H_{rd} = [H_r, H_{res}] \in \mathbb{R}^{n \times k_p}$ {Concatenate reference + residual}
return $H_{rd}$
```

Algorithm 3 Chained Reverse Distillation

```
Require: Dataset $\mathcal{D}$, model hierarchy $\{M_1, \ldots, M_m\}$
Ensure: Decomposition components $\{\mathbf{W}_i, \mathbf{V}_i\}_{i=2}^m$
Initialize: $H_{acc}^{(1)} = M_1(\mathcal{D})$, $k_{acc}^{(1)} = k_1$
for $i = 2$ to $m$ do
    $H_i = M_i(\mathcal{D})$
    Learn predictor: $\mathbf{W}_i = \arg\min_{\mathbf{W}} \|H_i - H_{acc}^{(i-1)}\mathbf{W}\|_F^2$
    Compute residuals: $R_i = H_i - H_{acc}^{(i-1)}\mathbf{W}_i$
    Apply SVD: $R_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$
    Select components: $\mathbf{V}_i = \mathbf{V}_i[:, 1 : (k_i - k_{acc}^{(i-1)})]$
    Update: $H_{acc}^{(i)} = [H_{acc}^{(i-1)}, R_i\mathbf{V}_i]$
    Update: $k_{acc}^{(i)} = k_i$
end for
return $\{\mathbf{W}_i, \mathbf{V}_i\}_{i=2}^m$
```

Theoretical Analysis Let $\mathcal{M}_r \subset \mathbb{R}^{k_r}$ be the manifold spanned by embeddings of M_r. To enable scalability and flexibility in our representation space, it is desirable to enforce the Matryoshka property on our embeddings. Thus, we consider the set of all k_p-dimensional representations that preserve M_r's embeddings in their first k_r coordinates:

$$\mathcal{C}_r = \{[H_r, X] : H_r \in \mathcal{M}_r, X \in \mathbb{R}^{L \times (k_p - k_r)}\}$$

Our decomposition $H_{rd} = [H_r, H_{res}]$ minimizes reconstruction error within this constrained space:

Theorem 1 (Optimal Constrained Approximation)**.** *Let $\tilde{H}_p \in \mathbb{R}^{L \times k_p}$ and $\tilde{H}_r \in \mathbb{R}^{L \times k_r}$ be stacked representations from models M_p and M_r respectively, where $r < p$. Among all representations of the form $[\tilde{H}_r, X]$ where $X \in \mathbb{R}^{L \times (k_p - k_r)}$, the representation $H_{rd} = [\tilde{H}_r, H_{res}]$ with H_{res} derived from the top $(k_p - k_r)$ singular vectors of the residual $\mathbf{R} = \tilde{H}_p - \tilde{H}_r \mathbf{W}^*$ minimizes:*

$$\min_{A \in \mathbb{R}^{k_p \times k_p}} \|\tilde{H}_p - [\tilde{H}_r, X]A\|_F^2$$

Proof. The proof directly follows from the Eckart-Young theorem. The optimal linear predictor is $\mathbf{W}^* = (\tilde{H}_r^T \tilde{H}_r)^{-1} \tilde{H}_r^T \tilde{H}_p$, minimizing the reconstruction error over all $\mathbf{W} \in \mathbb{R}^{k_r \times k_p}$. The residual $\mathbf{R} = \tilde{H}_p - \tilde{H}_r \mathbf{W}^*$ contains information orthogonal to $\tilde{H}_r$. For $\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the optimal rank-$(k_p - k_r)$ approximation uses the top $(k_p - k_r)$ singular vectors. □

3 Experiments

3.1 Initial Exploration of Model Chain Configuration

We began by investigating the optimal chaining of small models into larger models. For three ProteinGym DMS datasets, we evaluated a range of chain configurations. Let $K = \{k_0, k_1, \ldots, k_n\}$ denote the $n + 1$ model sizes from a model family (for ESM-2: $K = \{8M, 35M, 150M, 650M, 3B, 15B\}$). For a target embedding k_t with $t \in [0, n]$, the chain configuration was defined as follows:

1. for each $k_i \in [0, n]$ with $i < t$, a direct chain $k_i \rightarrow k_t$
2. longest chain: $k_0 \rightarrow k_0 \rightarrow \cdots \rightarrow k_t$

As shown in Table 1, a consistent trend emerged in which longer incremental chains yielded improved performance. Consequently, we concentrated our comprehensive experiments on the results obtained from reverse distillation of the two largest models.

Table 1: **Progressive chain vs. direct chain.** Our approach supports reverse distillation of any smaller model into any larger model. However, we find empirically that the best performance comes from progressively distilling up a "chain" of models. This progressive chain is what we refer to as rd.650M, rd.3B, and rd.15B in the rest of the manuscript.

ESM Models	**ARGR_ECOLI** **Tsuboyama_2023_1AOY**	**DN7A_SACS2** **Tsuboyama_2023_1JIC**	**ILF3_HUMAN** **Tsuboyama_2023_2L33**
8M	0.771	0.746	0.670
35M	0.767	0.806	0.692
rd: 8M→35M	0.776	0.793	0.701
150M	0.799	0.786	0.760
rd: 35M→150M	0.811	0.791	0.772
rd: 8M→150M	0.820	0.792	0.779
650M	0.834	0.868	0.712
rd: 8M→650M	0.849	0.878	0.765
rd: 35M→650M	0.835	**0.881**	0.759
rd: 150M→650M	0.845	0.866	0.751
rd: 8M→35→150→650M (rd.650M)	**0.858**	0.867	**0.786**
3B	0.845	0.880	0.749
rd: 8M→3B	0.852	**0.898**	0.780
rd: 35M→3B	0.844	0.894	0.777
rd: 150M→3B	0.853	0.886	0.775
rd: 650M→3B	0.859	0.880	0.751
rd: 8→35→150→650→3B (rd.3B)	**0.873**	0.890	**0.801**

In the rest of the paper, we denote the chain $k_{8M} \rightarrow \cdots \rightarrow k_{650M}$ as **rd.650**, the chain $k_{8M} \rightarrow \cdots \rightarrow k_{3B}$ as **rd.3B**, and the chain $k_{8M} \rightarrow \cdots \rightarrow k_{15B}$ as **rd.15B**.

3.2 ProteinGym DMS Analysis

For a comprehensive analysis, we obtained ProteinGym datasets with at least one double- or multi-mutation variant. We excluded datasets with fewer than 100 single-mutation variants, to ensure that our evaluation estimates were reliable. Given an embedding scheme, for each protein in the dataset, we loaded the embedding of the wild-type sequence and the embeddings of the mutated sequence. For each mutation, we computed the embedding difference vector between the mutated sequence and the corresponding wild-type sequence at the mutated position, feeding it into a ridge regression classifier. For variants with multiple mutations, we first average the differences across all mutated positions. We fitted the ridge regression on 80%

of the single-mutational variants using leave-one-out cross-validation. The fitted model was used to predict for all multiple-mutants and the remaining single-mutant cases. Note that since rd.650M (rd.3B) is a prefix of rd.3B (rd.15B) by construction (the first 1280 (2560) dimensions are identical), any cases where rd.3B (rd.15B) underperforms rd.650M (rd.3B) likely reflect ridge regression artifacts rather than representational limitations.

Table 2: **Reverse Distillation restores scaling on ProteinGym benchmarks.** We show that not only do rd.650M, rd.3B, and rd.15B consistently outperform their baseline counterparts, but that reverse distillation preserves expected scaling, i.e. larger models more frequently outperform smaller models.

	# ProteinGym DMS Datasets	**% ProteinGym DMS Datasets where one model outperforms another**						
		rd.650M > 650M	**rd.3B > 3B**	**rd.15B > 15B**	**3B > 650M**	**15B > 3B**	**rd.3B > rd.650M**	**rd.15B > rd.3B**
ProteinGym DMS Datasets with 1 and 2 mutations								
1 mut	28	50.00%	75.00%	64.29%	60.71%	89.29%	92.86%	92.86%
2 mut	28	50.00%	53.57%	57.14%	57.14%	82.14%	64.29%	78.57%
ProteinGym DMS Datasets with >2 mutations								
1 mut	6	33.33%	16.67%	83.33%	100.00%	16.67%	100.00%	83.33%
2 mut	6	66.66%	66.66%	83.33%	16.67%	100.00%	66.67%	100.00%
3 mut	6	66.66%	66.66%	83.33%	16.67%	100.00%	66.67%	100.00%
4 mut	6	50.66%	66.66%	66.66%	33.33%	66.67%	66.67%	83.33%

For each ProteinGym DMS dataset, we computed the Spearman correlation between our predicted scores and the ground truth; we followed the ProteinGym creators in our choice of the Spearman metric. In Tables 2 and 3, we report an estimated per-dataset measure of improvement, asking "in how many datasets does model M_1 outperfom M_2" along with the mean and standard deviations of these correlations for the ESM-2 family of models as well as their reverse-distilled version.

Table 3: **Spearman correlation of predicted mutational effect on ProteinGym benchmarks.** rd.15B achieves the strongest performance out of any model tested, and the reverse distilled models generally outperform their baseline counterparts.

	# ProteinGym DMS Datasets	**Test Spearman correlation (mean ± std)**					
		650M	**rd.650M**	**3B**	**rd.3B**	**15B**	**rd.15B**
ProteinGym DMS Datasets with 1 and 2 mutations							
1 mut	28	0.881 (± 0.040)	**0.884 (± 0.038)**	0.884 (± 0.043)	**0.893 (± 0.039)**	0.899 (± 0.036)	**0.904 (± 0.037)**
2 mut	28	0.677 (± 0.140)	**0.678 (± 0.133)**	0.682 (± 0.128)	**0.697 (± 0.115)**	0.714 (± 0.115)	**0.720 (± 0.120)**
ProteinGym DMS Datasets with >2 mutations							
1 mut	6	**0.458 (± 0.302)**	0.457 (± 0.307)	**0.501 (± 0.304)**	0.475 (± 0.311)	0.480 (± 0.321)	**0.495 (± 0.312)**
2 mut	6	**0.532 (± 0.227)**	0.529 (± 0.229)	0.527 (± 0.236)	**0.553 (± 0.235)**	0.560 (± 0.242)	**0.591 (± 0.240)**
3 mut	6	**0.510 (± 0.155)**	0.500 (± 0.160)	0.494 (± 0.166)	**0.524 (± 0.168)**	0.524 (± 0.183)	**0.569 (± 0.175)**
4 mut	6	**0.497 (± 0.137)**	0.466 (± 0.134)	0.462 (± 0.145)	**0.485 (± 0.139)**	0.498 (± 0.160)	**0.540 (± 0.131)**

3.3 Additional Protein Property Prediction

We evaluated our reverse-distilled models on several downstream protein property prediction tasks where the prediction task directly corresponded to protein structural, functional, and dynamic features. We utilized the 3-class secondary structure prediction (SSP Q3), 8-class secondary structure prediction (SSP Q8), metal ion binding (MIB), structural fold prediction (FOLD), and localization prediction (LOC) benchmarks from Biomap Research, and R2/R1 prediction from RelaxDB (Wayment-Steele et al., 2025). Training was performed analogously to the previous setting. The reverse-distilled models frequently outperformed the base models (Table 4) and demonstrated consistent scaling, with rd.3B always outperforming rd.650M and the baseline ESM models.

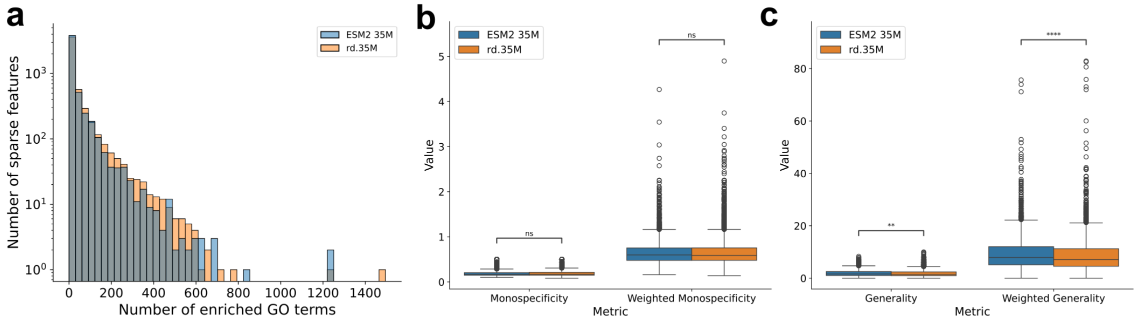

Figure 2: **Reverse Distillation embeddings capture more specific GO terms.** **(a)**SAE features from the rd.35M model are enriched for more GO terms than those from the base model. **(b)** The sets of GO terms for each model are equally compact, measured by pairwise shortest path on the GO tree. **(c)** The sets of GO terms for rd.35M are significantly more specific, measured by the depth of the pairwise least common ancestor on the GO tree.

These results provide indirect support for our hypothesis about hierarchical feature organization. Tasks requiring specialized or higher-order features—metal ion binding and protein dynamics (R2/R1)—show substantially larger gains from scaling (MIB: +0.066; R2/R1: +0.026). Notably, for R2/R1, the gap between 3B and rd.3B (0.369 to 0.425) exceeds the gap between 650M and 3B (0.343 to 0.369), suggesting that disentanglement via reverse distillation is particularly beneficial for tasks that rely on specialized features.

3.4 Probing embeddings with Sparse Autoencoders

Finally, we sought to explore whether our reverse-distilled embeddings were capturing more “specialized” biological features by training sparse autoencoders (SAE). We followed the analysis of Gujral et al. (2025), training an SAE on embeddings from both the base ESM2 35M and our rd.35M model. Using a set of 18,142 proteins from the Uniprot database, we identified the proteins significantly associated with each sparse feature, then performed a GO enrichment analysis using annotations for the proteins (Ashburner et al., 2000) to identify the set of GO terms associated with each sparse feature. We found that the SAE trained on rd.35M embeddings contained more enriched GO terms (40 GO terms for the average rd.35M SAE feature, vs. 32 for the average ESM2 35M SAE feature) indicating that these embeddings capture more functional features than those from the base model (Figure 2a).

In addition, the DAG relationship between GO terms allows us to probe the relationships within each functional set. Following Gujral et al. (2025) we compute “monospecificity,” the inverse of average shortest pairwise distance between GO terms in the same set, and “generality,” the depth of the least common ancestor (LCA) of each pair of terms in the set. A high monospecificity means that SAE features capture a more compact set of features, while a high generality means that SAE features capture broader biological function. While rd.35M and ESM2 35M embeddings have similar levels of monspecificity (Figure 2b), rd.35M embeddings are significantly less general, lending support to our hypothesis that reverse distillation helps to extract more “specialized” biological features.

3.5 Inference time

On an Nvidia A6000 GPU, embedding a protein sequence (mean length = 536) took 0.09s and 0.249s for the ESM-2 650M and 3B models respectively. Even though rd.650M involves four ESM model-invocations (8M, 35M, 150M and 650M) it only took 1.69x(=0.152s) the time as the smaller models have faster inference. Similarly, rd.3B makes five model-invocations but took only 1.53x (=0.380s) the time compared to baseline ESM-2 3B, and the six model-invocations of rd.15B take only 1.70x as long as the base model.

Thus reverse distillation does not have a prohibitive inference overhead. We also note the prefix structure of the embeddings that enhances reusability.

Table 4: **Reverse distillation of ESM-2 improves performance on downstream protein property prediction tasks.** We evaluate ESM2 650M, 3B and their corresponding reverse-distilled versions on data sets from Biomap Research and two protein-dynamics benchmarks. We find across all data sets that rd.3B achieves the strongest performance.

Dataset	**650M**	**rd.650M**	**3B**	**rd.3B**
SSP Q3	0.742	0.741	0.751	**0.762**
SSP Q8	0.691	0.691	0.703	**0.714**
MIB	0.511	0.506	**0.577**	**0.577**
FOLD	0.638	0.655	0.658	**0.661**
LOC	0.695	0.690	0.691	**0.707**
R2/R1	0.343	0.405	0.369	**0.425**

4 Related Work

ProteinGym analyses argue that performance gains plateau around the 1–4B range (Notin, 2025) and that hybrids leveraging MSAs/structure often win on zero-shot fitness, indicating a mismatch between current pretraining objectives and many downstream tasks. Li et al. (2024) show that improvements with scale are largely task-dependent—structure prediction aligns with pretraining, but many other tasks draw mainly on features learned early, so linear probes on large, mixed representations struggle to isolate task-relevant signal. Zhang et al. (2024) introduced the categorical Jacobian and estimated that an ESM-2 3B model delivers contact-recovery signals comparable to the 15B variant, reinforcing diminishing returns past a few billion parameters. Consistent with this, Vieira et al. (2025) find medium-sized models (approx. 600–650M) perform competitively in realistic transfer settings. Recently, Hou et al. (2025) link downstream variant-effect accuracy to an intermediate model perplexity band (roughly “medium” pseudo-/perplexity): models that are too uncertain or too certain both degrade discrimination, which helps explain why very large models can underperform.

Several works have analyzed the structure and redundancy of protein representations. Lu et al. (2025) show that embeddings can be significantly compressed along both sequence length and feature dimensions without losing predictive power. Devkota et al. (2024) provide complementary evidence for representational redundancy through alternative compression schemes. These compression results suggest that large PLMs learn representations with substantial redundancy, motivating approaches that can selectively extract and combine the most informative components across model scales.

Traditional knowledge distillation (Hinton et al., 2015) focuses on transferring knowledge from large teacher models to smaller student models. However, our approach is fundamentally different: instead of compressing a large model into a small one, we systematically decompose large representations to understand and leverage the unique contributions of each model scale. Recent work on model combination (Wortsman et al., 2022) and ensemble methods provides related techniques, but typically lacks the theoretical guarantees and systematic subspace analysis that our approach provides. Our approach is also related to, but distinct from, methods for continual learning like o-LoRA (Wang et al., 2023) or Adaptive SVD (Nayak et al., 2025). While these focus on maintaining model performance with adaptation to new tasks by ensuring orthogonal subspaces of weights are updated, our approach provides a task-agnostic framework for extracting representations from multi-scale model families by maximizing residual information gain.

We note that this approach is similar to but distinct from methods that inspect individual attention heads in large models (“BERTology”) or seek feature interpretability via sparse autoencoders (SAEs). Attention-head analyses probe what heads encode without decomposing the representation itself (Rogers et al., 2020; Clark et al., 2019; Vig et al., 2021), while SAE studies in proteins aim to enumerate latent features explicitly, e.g., Gujral et al. (2025) or Adams et al. (2025). However, mapping SAE latents to task-relevant biological features generally demands heavy manual annotation; our method avoids this by operating implicitly, without pre-defining or cataloging features.

5 Conclusion

We introduce reverse distillation, a principled method for addressing scaling challenges in protein language models (PLMs) through structured subspace decomposition. This approach not only provides theoretical guarantees for the quality of approximation but also offers practical, parameter-efficient implementation benefits. By shifting the focus from “when do large models help?” to “how can we systematically extract and combine the unique contributions of models at different scales?”, our work opens up new research avenues in representation analysis and lays the groundwork for more effective PLM scaling strategies.

The success of our linear decomposition method indicates that many scaling challenges in PLMs result from inefficient use of representational capacity rather than fundamental limits in model expressiveness. By providing structured ways to combine multi-scale representations, we enable better utilization of computational resources while highlighting more efficient scaling paradigms.

Limitations and Future Work To further advance this research, future work will explore several key directions. We plan to investigate non-linear scaling methods to capture more complex relationships between model representations, moving beyond our current linear approach. Initial explorations of non-linear methods show improved mapping from low- to high-dimensional embeddings ($8M \rightarrow 35M$ $R^2 = 0.528$ vs. 0.422, $650M \rightarrow 3B$ $R^2 = 0.400$ vs. 0.261), showing promise to further extract unique features of the larger models. Additionally, we will explore enhanced dimensionality reduction techniques. For instance, we could first reduce the dimension space of the ESM2-8M model, a strategy that would also be valuable for other models using the same embedding dimension for different model sizes, allowing our approach to remain effective across various architectures. A non-linear dimensionality reduction such as UMAP (McInnes et al., 2018) could more effectively disentangle the residual features.

We also plan to use parameter-efficient fine-tuning (PEFT) methods such as LoRA to finetune a large model, directly producing the reverse-distilled embeddings at the last layer. This would facilitate generative use-cases of the model and enable likelihood- and logit-based probing methods. While our current approach introduces only a small-constant linear slowdown over the baseline methods, this approach would also provide a more efficient pipeline for downstream applications by requiring only a single forward pass, thereby completely erasing any slowdown. Additionally, we will explore the effect of reverse distillation on other biological foundation models beyond ESM-2 to test the generalizability of our approach. This includes investigating other PLMs (such as autoregressive models like Progen) as well as models for genomics and drug discovery. Furthermore, we will explore the application of reverse distillation to foundation models outside the biological domain, such as those for natural language processing or computer vision. The core principle of our method—leveraging a smaller model to systematically extract and combine the contributions of larger models—might be a fundamental property of models in general, not just those in biology. This broader exploration could reveal new insights into model scaling and representation learning that are applicable across diverse scientific and technological domains, highlighting the universal potential of our approach.

6 Reproducibility Statement

To facilitate easy verification and replication of our results, we have modularized our source code, added README documentation and provided information on how to train the reverse distillation models. Anonymized code is available as Supplementary Material in a zip format.

References

Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. In ICML, 2025. URL https://openreview.net/forum?id=zdOGBRQEbz.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. Nature genetics, 25(1):25–29, 2000.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention. In Proceedings of BlackboxNLP, 2019.

Kapil Devkota, Daichi Shonai, Joey Mao, Young Su Ko, Wei Wang, Scott Soderling, and Rohit Singh. Miniaturizing, modifying, and magnifying nature's proteins with raygun. biorxiv, pp. 2024–08, 2024.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. Nature communications, 13(1):4348, 2022.

Onkar Gujral, Mihir Bafna, Eric Alm, and Bonnie Berger. Sparse autoencoders uncover biologically interpretable features in protein language model representations. PNAS, 2025. doi: 10.1073/pnas.2506316122.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. arXiv preprint arXiv:2203.15556, 2022.

Chao Hou, Di Liu, Aziz Zafar, and Yufeng Shen. Understanding language model scaling on protein fitness prediction. bioRxiv, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning, 2024. URL https://arxiv.org/abs/2205.13147.

Francesca-Zhoufan Li, Ava P Amini, Yisong Yue, Kevin K Yang, and Alex X Lu. Feature reuse and scaling: Understanding transfer learning with protein language models. bioRxiv, pp. 2024–02, 2024.

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.

Amy X Lu, Wilson Yan, Kevin K Yang, Vladimir Gligorijevic, Kyunghyun Cho, Pieter Abbeel, Richard Bonneau, and Nathan C Frey. Tokenized and continuous embedding compressions of protein sequence and structure. Patterns, 6(6), 2025.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Hao Wang, Aldo Pareja, Oleg Silkin, Mustafa Eyceoz, et al. Sculpting subspaces: Constrained full fine-tuning in llms for continual learning. arXiv preprint arXiv:2504.07097, 2025.

Pascal Notin. Have we hit the scaling wall for protein language models? https://pascalnotin.substack.com/p/have-we-hit-the-scaling-wall-for, 2025. Accessed: 2025-09-25.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. Transactions of the Association for Computational Linguistics, 2020. doi: 10.1162/tacl_a_00349.

Luiz C Vieira, Morgan L Handojo, and Claus O Wilke. Medium-sized protein language models perform well at transfer learning on realistic datasets: Lc vieira et al. Scientific Reports, 15(1):21400, 2025.

Jesse Vig, Martin Jaggi, and et al. Bertology meets biology: Interpreting attention in protein language models. In ICLR, 2021.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 10658–10671, 2023.

Hannah K Wayment-Steele, Gina El Nesr, Ramith Hettiarachchi, Hasindu Kariyawasam, Sergey Ovchinnikov, and Dorothee Kern. Learning millisecond protein dynamics from what is missing in nmr spectra. bioRxiv, pp. 2025–03, 2025.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In International conference on machine learning, pp. 23965–23998. PMLR, 2022.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. Science, 379(6639):1358–1363, 2023.

Zhidian Zhang, Hannah K Wayment-Steele, Garyk Brixi, Haobo Wang, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. Proceedings of the National Academy of Sciences, 121(45):e2406285121, 2024.