PROTOCIL: CLASS-INCREMENTAL LEARNING FOR PROTEIN REPRESENTATIONS WITH ADAPTIVE PROTOTYPES

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein function annotations continually expand as new enzyme classes are discovered, requiring models to incorporate new classes without retraining from scratch. Yet class-incremental learning for protein function prediction remains underexplored, especially under frozen protein encoders with highly heterogeneous embedding geometries. We present the first systematic benchmark for protein class-incremental learning, covering two enzyme datasets, five frozen protein encoders, and eleven representative CIL baselines. Our benchmark reveals that no fixed-granularity approach, whether shared covariance, random projection, or learned subspace, scales reliably across encoders and class counts. We propose ProtoCIL, a training-free prototype method that adaptively learns group-wise covariance structure over frozen protein embeddings for Mahalanobis classification. ProtoCIL derives its target group count from intrinsic rank and sample size, then refines groups incrementally through validation-based splits as new classes arrive. Across all encoder-dataset configurations, ProtoCIL achieves the highest average accuracy, outperforming the strongest baseline by 8.8% on Enzyme Reaction and 18.5% on EC Number, while maintaining average forgetting below 5%. These results demonstrate that adaptive group-wise covariance sharing provides an effective principle for protein class-incremental learning. Code is provided for review and will be released upon acceptance.

1 INTRODUCTION

Why protein CIL? The label space of protein function annotation keeps expanding. The IUBMB enzyme list alone has introduced nearly 3,000 new EC numbers since 2010, about 175 per year (McDonald et al., 2009), and functional databases such as Swiss-Prot incorporate these new classes with every release (Consortium, 2023). A deployed function prediction model therefore faces a recurring need to recognize newly defined enzyme classes. Retraining from scratch after each release is costly, as its cost grows with the size of the accumulated database (Masana et al., 2023; Patil et al., 2026), and may be infeasible when earlier training data are proprietary or no longer accessible. Class-incremental continual learning (CIL) offers a natural solution: the model absorbs new classes progressively while retaining discriminative ability on previously learned ones (Masana et al., 2023). Despite extensive study of CIL in computer vision (Rebuffi et al., 2017; Buzzega et al., 2020), no systematic investigation of class-incremental learning exists for protein function prediction.

Challenges of protein CIL. Directly transferring CIL methods from vision to proteins faces three domain-specific challenges. First, the protein domain features multiple pretrained encoders spanning fundamentally different architectural paradigms: sequence-based protein language models (Lin et al., 2023; Elnaggar et al., 2022), structure-based graph neural networks (Zhang et al., 2023b), and fusion models that combine both modalities (Su et al., 2024). These encoders produce embedding spaces with different dimensionalities, scales, and geometric properties, causing the same CIL strategy to yield dramatically different performance across encoders. Second, existing CIL methods have hyperparameters that implicitly depend on the geometry of the embedding space. A regularization strength (Kirkpatrick et al., 2017) or replay buffer size (Chaudhry et al., 2019) that works well for one encoder may cause classifier collapse on another, requiring per-encoder tuning that defeats the purpose of a plug-and-play continual learning system. Third, proteins possess a natural hierarchical functional

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.

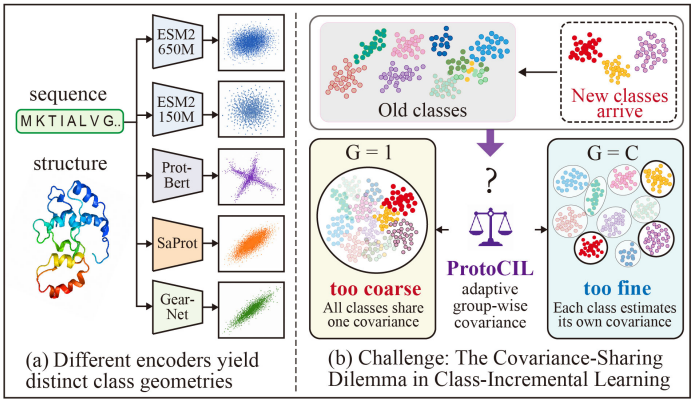


Figure 1: Motivation of ProtoCIL. Frozen protein encoders induce heterogeneous class geometries, making covariance modeling difficult as new classes arrive.

taxonomy (e.g., the four-level EC number hierarchy (Webb, 1992)), where enzymes within the same functional family share catalytic mechanisms and active-site geometry. This domain-specific structure could provide biological priors for CIL strategy design, but current methods have not exploited it.

This paper makes three contributions:

First benchmark for protein CIL. We establish a class-incremental learning benchmark for protein function prediction under the frozen-encoder regime, covering two datasets, five frozen encoders spanning three architectural paradigms, and eleven baseline CIL methods across 110 evaluation configurations. The benchmark reveals that gradient-based CIL methods lose their stability advantage under frozen encoders, and that no fixed-granularity approach scales reliably to thousands of classes.

ProtoCIL: adaptive covariance grouping for protein CIL. We propose ProtoCIL, a training-free prototype-based method for adaptive Mahalanobis classification over frozen protein embeddings, without per-encoder tuning. ProtoCIL partitions classes into covariance groups whose number is derived in closed form from shrinkage estimation theory (Ledoit & Wolf, 2004; Chen et al., 2010), and grows incrementally as new classes arrive. The grouping naturally aligns with the structure of protein embedding spaces, where functionally related enzymes share similar variance patterns.

State-of-the-art performance across encoders and scales. ProtoCIL achieves the best average accuracy across 10 encoder-dataset configurations, outperforming the strongest baseline by 8.8% on Enzyme Reaction and by 18.5% on EC Number, with average forgetting below 5%. On EC Number it approaches the Joint MLP reference without any backpropagation or replay buffer.

2 Related Work

Protein representation learning has rapidly shifted from task-specific supervised models to general-purpose pretrained encoders. Sequence-based protein language models such as TAPE, ESM, ESM2, and ProtTrans show that large-scale self-supervised pretraining can encode structural and functional signals from protein sequences (Rao et al., 2019; Rives et al., 2021; Lin et al., 2023; Elnaggar et al., 2022), while structure-aware models such as GearNet, SaProt, and PST further incorporate geometric or structural information into protein representations (Zhang et al., 2023b; Su et al., 2024; Hartout et al., 2025). These representations are widely used for protein annotation tasks, including enzyme function prediction and EC-number classification. However, protein annotation is typically evaluated under fixed-label settings in existing benchmarks (Rao et al., 2019; Yu et al., 2023), and therefore does not explicitly address the continual emergence of new functional classes. In parallel, class-incremental learning studies how to incorporate new classes while preserving old-class performance (Masana et al., 2023), with classical approaches relying on regularization, replay, or exemplar memory (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Chaudhry et al., 2019; Buzzega et al., 2020). Recent pretrained-model-based CIL methods shift the focus from representation learning to adaptation over strong pretrained features, including prompt tuning, classifier alignment, random projections, and ensemble-based adaptation (Wang et al., 2022; Smith et al., 2023; Zhang et al., 2023a;

McDonnell et al., 2023; Zhou et al., 2024c; Sun et al., 2025). Yet these methods are primarily designed for vision representations rather than protein embeddings. Prototype-based classifiers offer a natural way to add new classes without retraining encoders (Mensink et al., 2013), but reliable covariance modeling remains difficult in high-dimensional protein embedding spaces, where class-wise estimates are sample-limited and global covariance can ignore heterogeneous functional geometry. This leaves open the problem of class-incremental learning over frozen pretrained protein representations under evolving functional label spaces. See Appendix A for details.

3 PROTEIN CIL BENCHMARK

Despite extensive study of class-incremental learning in computer vision, systematic CIL evaluation remains largely unexplored in protein science. This gap is important because protein encoders differ substantially in architecture, dimensionality, and intrinsic geometry, making it unclear which CIL paradigms can handle such heterogeneity. We therefore construct a benchmark with eleven representative CIL baselines, two enzyme datasets, and five frozen protein encoders, yielding 110 evaluation configurations.

Benchmark Construction We evaluate on two enzyme function prediction datasets that probe complementary failure modes. Enzyme Reaction (Hermosilla et al., 2021; Zhang et al., 2023b) provides a moderate-scale setting with 348 classes and approximately 43 samples per class, where shared-covariance prototype methods have sufficient data to estimate reliable statistics. EC Number (Yu et al., 2023) drastically raises the difficulty: 4,788 classes with a mean of only 15 samples per class creates precisely the sample-starved regime where covariance estimation collapses. This combination tests whether any single method can maintain reliable performance across both regimes. Both datasets are originally multi-label, but most proteins carry a single enzymatic function (over 88% in every split). Following the standard single-label CIL setting, we assign each protein one label, taking the argmax of the label vector for Enzyme Reaction and the first annotation for EC Number, and exclude samples without matched structures for the structure-dependent encoders.

We include five frozen protein encoders with different geometric properties: sequence-based models ESM2-650M (d=1280), ESM2-150M (d=640) (Lin et al., 2023), and ProtBert (d=1024) (Elnaggar et al., 2022); structure-based GearNet-Edge (d=3072) (Zhang et al., 2023b); and sequence-structure fusion model SaProt (d=1280) (Su et al., 2024). All encoders remain frozen throughout training. This frozen-encoder regime is now standard practice in CIL methods built on pretrained backbones (Zhou et al., 2024b;a): it isolates the effect of the CIL strategy from encoder adaptation, preserves the quality of pretrained representations, and avoids the computational overhead of per-task fine-tuning.

We compare against baselines spanning regularization (EWC (Kirkpatrick et al., 2017)), replay (Replay (Chaudhry et al., 2019), iCaRL (Rebuffi et al., 2017), DER++ (Buzzega et al., 2020), RER (Wang et al., 2024)), projection (RanPAC (McDonnell et al., 2023)), subspace (EASE (Zhou et al., 2024c)), and prototype-based approaches (NCM (Mensink et al., 2013) and FeCAM (Goswami et al., 2023)), plus Naive (lower bound) and Joint MLP (joint training reference). For FeCAM we report both the configuration of the original paper, FeCAM (orig.), and a ridge-regularized shared covariance, FeCAM (ridge). All hyperparameters are fixed across every encoder-dataset configuration to avoid per-setting tuning. To verify that this setting does not disadvantage the baselines, we also tune key hyperparameters of RanPAC, FeCAM (orig.), and DER++ per encoder on a held-out validation split (Appendix D).

Classes are partitioned into seven phases by EC first digit, each representing a distinct catalytic mechanism. Phases are also largely disjoint in sequence space: only 2.9% of Enzyme Reaction and 1.4% of EC Number proteins share more than 50% sequence identity with any protein from a different phase (Appendix B.5). This defines our main biologically structured phase construction, while additional random-class, finer-EC, mixed-hierarchy, and temporal protocols are evaluated in the experiments. Phase order is shuffled per random seed, except under the temporal protocol, which follows chronological order, and all methods within the same seed observe the same order. After each phase, accuracy is evaluated on test samples from all observed phases, yielding an accuracy matrix $R \in \mathbb{R}^{T \times T}$. From this matrix we compute three standard metrics (Masana et al., 2023): Average accuracy measures final retained performance (Lopez-Paz & Ranzato, 2017), forgetting measures degradation from the best historical performance (Chaudhry et al., 2018), and plasticity measures

new-class acquisition at each phase:

$$\text{AvgAcc} = \frac{1}{T}\sum_{j=0}^{T-1} R[T{-}1][j], \tag{1a}$$

$$\text{Fgt} = \frac{1}{T-1}\sum_{j=0}^{T-2}\left(\max_{i\in[j,T-2]} R[i][j] - R[T{-}1][j]\right), \tag{1b}$$

$$\text{Plast} = \frac{1}{T}\sum_{i=0}^{T-1} R[i][i]. \tag{1c}$$

Dataset statistics, encoder extraction details, and baseline configurations are provided in Appendix B. The benchmark reveals two patterns that collectively motivate our approach:

Stability is the bottleneck, and gradient-based mechanisms cannot solve it. Under frozen encoders, all gradient-based methods achieve near-identical plasticity exceeding 94% on EC Number for sequence-based encoders, yet their average accuracies span a large range. Since the frozen encoder provides representations expressive enough to absorb new classes, the differentiating factor is how well each method preserves old knowledge. However, gradient-based stability mechanisms cannot address this gap: RER (Wang et al., 2024), which monitors gradient conflicts to adaptively adjust replay strength, closely matches vanilla Replay across all configurations. When the encoder is frozen, gradients flow only through a small MLP head, making gradient-conflict signals less informative for stability. This suggests that effective stability must come from the embedding geometry.

No single non-adaptive strategy scales to thousands of classes. Prototype-based classification with Mahalanobis distance operates directly on embedding geometry without backpropagation, making it well suited to the frozen-encoder setting. FeCAM (Goswami et al., 2023) confirms this at moderate scale on Enzyme Reaction, but its single shared covariance cannot represent the geometric diversity of thousands of classes on EC Number. RanPAC (McDonnell et al., 2023) avoids covariance estimation entirely via random projection, achieving strong accuracy on sequence-based encoders but collapsing on high-dimensional GearNet (d=3072). EASE (Zhou et al., 2024c) learns per-task subspaces but its adapter-based features lack the geometric modeling needed for fine-grained enzyme discrimination. That is: no fixed-granularity approach, whether a single covariance, a random projection, or a learned subspace, can adapt to the heterogeneous geometry of diverse protein encoders and class scales. An encoder-agnostic approach that adapts to both geometry and scale is needed.

These two observations establish the design constraints for ProtoCIL: exploit embedding geometry rather than training dynamics for stability, and partition covariance structure adaptively rather than relying on a single shared estimate.

4 METHOD

4.1 PROBLEM FORMULATION

We consider class-incremental learning over a sequence of T phases $\mathcal{P} = \{P_0, P_1, \ldots, P_{T-1}\}$, where each phase P_t introduces a disjoint set of new classes. A frozen pretrained protein encoder f_θ maps each protein to a d-dimensional embedding $\mathbf{x} = f_\theta(\text{protein}) \in \mathbb{R}^d$. After observing phase P_t, the system must classify any query protein over all classes seen so far, $\mathcal{C}_{\leq t} = \bigcup_{i=0}^{t} P_i$, using frozen embeddings and maintained class statistics. Each functional class may contain only tens of training samples while d ranges from 640 to 3072 depending on the encoder. The core challenge is therefore to design a classifier that exploits the covariance structure of protein embeddings while remaining statistically well-posed under these sample constraints, and that can incorporate new classes without altering previously learned statistics.

4.2 INCREMENTAL GROUPING

To address the sample insufficiency described above, we organise classes into groups that share a single covariance model, and grow these groups incrementally as new phases arrive. For each class c

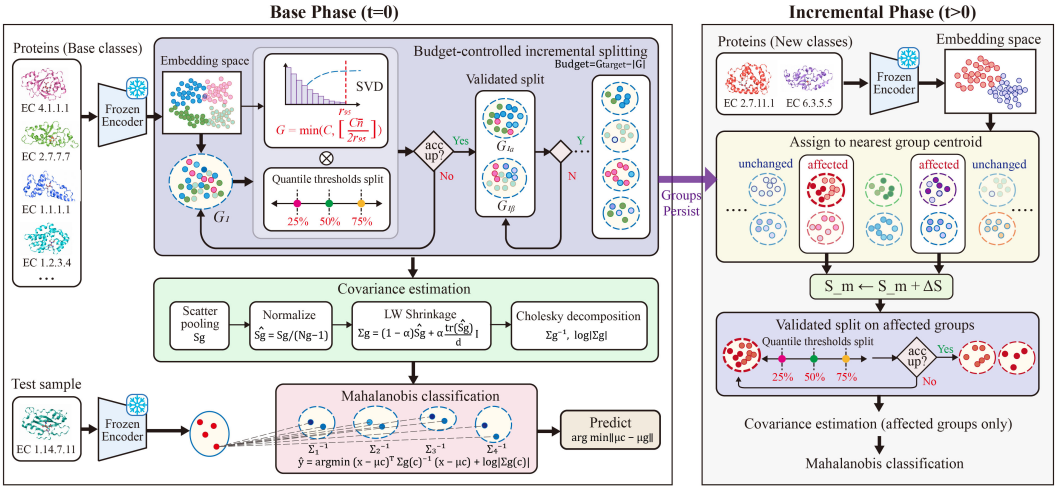


Figure 2: Overview of ProtoCIL Framework.

with n_c training embeddings $\{\mathbf{x}_i\}_{i=1}^{n_c}$, we compute a prototype (class mean) (Mensink et al., 2013; Rebuffi et al., 2017) and a within-class scatter matrix:

$$\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{x}_i, \qquad \mathbf{S}_c = \sum_{i=1}^{n_c}(\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^\top \tag{2}$$

where $\boldsymbol{\mu}_c \in \mathbb{R}^d$ is the prototype and $\mathbf{S}_c \in \mathbb{R}^{d\times d}$ captures the spread of class c around its mean. With per-class samples far below d, class-wise covariances as used in FeCAM (Goswami et al., 2023) are unreliable, while a single shared covariance cannot capture thousands of heterogeneous classes. We therefore pool scatter matrices within each group, interpolating between these two extremes, as functionally related protein classes tend to share similar variance structure (Wang et al., 2025a; Yeung et al., 2023):

$$\mathbf{S}_g = \sum_{c\in g}\mathbf{S}_c \tag{3}$$

In the base phase (t=0), all classes are placed into a single group (G=1). Only a small portion of the function space has been observed and the data does not support finer partitioning. When a subsequent phase introduces new classes, each new class c is assigned to the group whose centroid (mean of member prototypes) is nearest in Euclidean distance:

$$g(c) = \arg\min_{g\in\mathcal{G}} \|\boldsymbol{\mu}_c - \bar{\boldsymbol{\mu}}_g\| \tag{4}$$

where $\bar{\boldsymbol{\mu}}_g = \frac{1}{|g|}\sum_{c'\in g}\boldsymbol{\mu}_{c'}$ is the centroid of group g and $|g|$ is the number of member classes. This nearest-centroid rule exploits the tendency of protein encoders to embed functionally related proteins nearby: a newly arriving oxidoreductase subclass, for instance, will naturally fall close to the group already containing other oxidoreductases. The group's scatter matrix is updated incrementally via $\mathbf{S}_g \leftarrow \mathbf{S}_g + \mathbf{S}_c$. Groups that did not receive new classes remain unchanged: their scatter matrices and all derived quantities are preserved, ensuring zero modification of previously learned statistics. This is important because covariance models established for well-characterised enzyme families in earlier phases should not be affected by unrelated classes arriving later.

4.3 Target Group Count and Splitting

The group covariance is estimated via the Ledoit–Wolf (LW) shrinkage estimator (Ledoit & Wolf, 2004), which regularises the normalised sample covariance $\hat{\mathbf{S}}_g = \mathbf{S}_g/(N_g - 1)$ (where N_g is the total number of samples in group g) by mixing it with an isotropic target:

$$\boldsymbol{\Sigma}_g = (1-\alpha)\,\hat{\mathbf{S}}_g + \alpha\,\frac{\mathrm{tr}(\hat{\mathbf{S}}_g)}{d}\,\mathbf{I} \tag{5}$$

The shrinkage intensity $\alpha \in [0, 1]$ is computed from the data via the Oracle Approximating Shrinkage (OAS) formula (Chen et al., 2010), requiring no manual tuning. When α is small, the sample covariance dominates and directional structure is preserved; as $\alpha \to 1$, the estimate collapses to a scaled identity that retains no directional information. We require $\alpha < 1/2$ so that the sample covariance contributes strictly more than the isotropic regulariser. When the data concentrates on a low-dimensional subspace, the OAS shrinkage intensity simplifies to (see Appendix C):

$$\alpha \approx \frac{r_{95}}{N_g} \tag{6}$$

where r_{95} is the *intrinsic rank* (Jolliffe, 2002): the smallest number of principal components explaining 95% of pooled within-class variance, computed via SVD of the class-centered embedding matrix from all seen classes:

$$r_{95} = \min\left\{ r : \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{j=1}^{d} \sigma_j^2} \geq 0.95 \right\} \tag{7}$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$ are the singular values of the centered embedding matrix. The value of r_{95} is encoder-dependent and is recomputed after each phase as new classes arrive.

Substituting equation 6 into the constraint $\alpha < 1/2$ yields $N_g > 2\, r_{95}$. With G groups over C classes of mean size $\bar{n}$, each group has on average $N_g = C\bar{n}/G$ samples:

$$G_{\text{target}} = \min\Big(C,\ \Big\lfloor \frac{C\,\bar{n}}{2\,r_{95}} \Big\rfloor\Big) \tag{8}$$

All quantities are computed from data and updated after each phase. As a secondary guard, if $C\bar{n}/G_{\text{target}} < d$, the sample covariance is rank-deficient and its null space carries no group-specific discriminative information. In this case we halve G_{target} repeatedly until $C\bar{n}/G_{\text{target}} \geq d$ or $G_{\text{target}}{=}1$. G_{target} is not imposed all at once but serves as a ceiling for the number of splits.

Split budget. Directly imposing G_{target} groups at once would require re-partitioning all existing classes, disrupting previously learned covariance models. Instead, we allow the number of groups to grow incrementally toward G_{target} through a split budget mechanism. After each phase, G_{target} is recomputed and the *split budget* is:

$$B = \max(0,\ G_{\text{target}} - |\mathcal{G}|) \tag{9}$$

where $|\mathcal{G}|$ is the current number of groups. Splits are only attempted when $B > 0$, so the number of groups grows toward but never exceeds G_{target}.

Validated multi-directional splitting. When a group contains at least M_{split} member classes, it is eligible for splitting:

$$M_{\text{split}} = 2\ \max(2, \lceil 2\, r_{95}/\bar{n} \rceil) \tag{10}$$

ensures both child groups retain enough samples for reliable covariance estimation (see Appendix C for derivation). For each eligible group, we center its member prototypes and compute a thin SVD. The top $k = \min(r_{95}, 5, |g|{-}1)$ principal directions are examined as candidate split axes. Along each direction, split points at the 25th, 50th, and 75th percentiles of projected prototype values are evaluated, yielding up to $3k$ candidate partitions per group. A held-out subset (10% of training samples, reserved per class before prototype computation) serves as the validation set. The candidate partition achieving the highest classification accuracy on this subset is selected, and the split is accepted only if accuracy exceeds the pre-split baseline. Newly created groups enter the queue for further splitting in breadth-first order until no beneficial splits remain or the budget is exhausted.

4.4 Mahalanobis Classification

After grouping and splitting are complete for a given phase, each group's covariance $\mathbf{\Sigma}_g$ is finalised via equation 5, and its inverse (precision matrix) $\mathbf{\Sigma}_g^{-1}$ and log-determinant $\log|\mathbf{\Sigma}_g|$ are obtained via Cholesky decomposition. A query embedding $\mathbf{x}$ is then classified over all seen classes:

$$\hat{y} = \arg\min_{c \in \mathcal{C}_{\leq t}} (\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{\Sigma}_{g(c)}^{-1} (\mathbf{x} - \boldsymbol{\mu}_c) + \log|\mathbf{\Sigma}_{g(c)}| \tag{11}$$

The quadratic form measures distance using the group's learned metric, where directions of high within-group variance are down-weighted and compact directions become more discriminative. The log-determinant normalises scores across groups with different covariance volumes, which is essential when groups contain varying numbers of classes or exhibit different variance scales.

Table 1: Benchmark results across two datasets and five encoders. Encoder-specific results are reported as mean ± std over 3 seeds. The Avg column is the arithmetic mean over the five encoder means. **Bold** and underline mark the best and second best result in each column (excluding Naive and Joint MLP). FeCAM (orig.) follows the configuration of the original paper; FeCAM (ridge) uses a ridge-regularized shared covariance.

		Enzyme Reaction (348 classes)						**EC Number (4788 classes)**					
Cat.	**Method**	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg
						Average Accuracy (%) ↑							
Lower	Naive	13.7±0.5	12.6±1.5	7.4±1.4	13.4±0.7	10.7±0.5	11.6	25.4±1.6	23.5±1.4	8.6±0.3	20.7±0.8	21.4±0.2	19.9
Joint	Joint MLP	73.9±0.8	69.1±0.8	52.1±1.1	71.2±0.4	65.5±0.3	66.4	99.5±0.2	99.5±0.1	68.4±0.7	99.4±0.0	99.5±0.0	93.3
Reg.	EWC	21.8±4.0	17.7±4.1	6.9±1.1	20.8±4.0	15.0±1.8	16.4	45.2±5.4	40.2±2.9	8.7±1.2	32.6±1.8	40.2±1.3	33.4
Replay	Replay	57.2±2.0	52.5±1.5	34.5±1.9	53.9±0.6	44.9±1.9	48.6	58.2±2.6	52.8±2.2	25.7±0.4	51.0±1.5	51.5±1.6	47.8
	iCaRL	62.5±0.6	56.4±0.8	36.9±0.8	60.1±0.8	51.2±0.4	53.4	59.5±1.7	52.9±2.1	24.0±1.9	50.6±1.0	52.2±0.9	47.8
	DER++	66.8±3.1	61.2±5.2	33.1±2.7	62.0±1.8	54.7±5.0	55.6	64.5±5.1	59.4±5.7	25.9±2.9	57.9±6.7	56.3±6.5	52.8
	RER	56.4±1.4	52.3±2.1	34.6±1.6	54.6±1.0	45.8±1.8	48.7	58.2±2.5	52.7±2.3	25.8±0.7	50.9±1.5	51.4±1.6	47.8
Subsp.	EASE	55.4±0.8	49.9±0.8	29.4±1.0	53.0±0.9	43.7±0.8	46.3	60.0±0.6	58.6±0.3	44.7±0.2	59.1±0.3	53.7±0.9	55.2
Proj.	RanPAC	74.3±0.0	70.5±0.0	21.8±0.0	73.5±0.0	65.8±0.0	61.2	82.2±0.0	79.9±0.0	26.1±0.0	84.7±0.0	79.0±0.0	70.4
Proto.	NCM	19.6±0.0	15.4±0.0	19.9±0.0	27.9±0.0	10.9±0.0	18.7	50.8±0.0	49.4±0.0	37.6±0.0	46.5±0.0	23.7±0.0	41.6
	FeCAM (ridge)	75.0±0.0	67.1±0.0	43.4±0.0	72.5±0.0	69.6±0.0	65.5	55.8±0.0	51.7±0.0	37.7±0.0	47.7±0.0	30.2±0.0	44.6
	FeCAM (orig.)	61.9±0.0	49.6±0.0	35.5±0.0	60.1±0.0	38.4±0.0	49.1	82.4±0.0	77.9±0.0	61.0±0.0	77.9±0.0	67.1±0.0	73.3
Ours	**ProtoCIL**	**81.6**±0.7	**76.8**±0.8	**62.7**±0.0	**78.4**±0.2	**71.8**±1.4	**74.3**	**96.4**±0.6	**96.0**±0.2	**82.9**±0.0	**95.1**±0.3	**88.8**±2.0	**91.8**
						Forgetting (%) ↓							
Lower	Naive	76.3±1.2	73.3±1.5	54.9±3.0	73.3±0.9	68.9±1.0	69.3	85.1±2.3	86.3±2.0	52.6±5.7	87.6±1.5	85.6±0.7	79.4
Reg.	EWC	66.5±4.6	66.4±3.9	53.9±2.4	65.2±4.7	62.5±1.8	62.9	62.0±6.5	66.7±3.2	52.2±4.5	73.7±1.4	63.7±2.2	63.7
Replay	Replay	23.0±2.5	24.7±2.1	24.2±1.3	25.3±0.2	28.0±1.2	25.0	47.5±3.4	52.8±3.7	36.1±5.0	53.6±2.8	51.7±2.8	48.3
	iCaRL	16.3±1.4	20.1±2.3	21.0±1.9	17.4±1.5	19.8±1.3	18.9	46.5±2.3	53.6±2.4	36.5±6.6	54.9±1.8	52.3±1.3	48.8
	DER++	12.7±1.0	15.3±3.1	23.2±1.9	15.3±1.1	16.0±1.1	16.5	39.7±5.5	44.4±5.8	26.4±4.5	44.7±6.7	45.3±5.2	40.1
	RER	24.5±1.9	25.7±2.9	22.6±0.6	24.4±1.0	27.2±0.8	24.9	47.5±3.3	52.9±3.7	35.5±5.1	53.7±2.8	51.7±2.8	48.3
Subsp.	EASE	11.0±2.0	9.9±3.4	12.7±2.0	11.6±1.7	9.8±1.7	11.0	8.0±0.7	6.7±0.3	5.8±1.0	6.6±1.2	9.1±1.0	7.2
Proj.	RanPAC	6.8±2.1	6.1±1.8	7.3±3.8	4.0±1.7	5.7±2.5	6.0	7.3±1.3	7.5±1.2	**3.3**±1.4	5.1±0.9	6.9±1.3	6.0
Proto.	NCM	3.4±1.2	3.1±1.0	8.4±2.1	6.1±2.2	**2.9**±0.7	4.8	2.7±1.0	2.6±1.0	3.9±1.5	3.6±1.0	**2.8**±1.3	3.1
	FeCAM (ridge)	5.8±1.6	8.0±0.9	9.8±2.6	5.6±0.7	6.1±1.2	7.1	3.4±1.1	3.4±1.3	3.9±1.4	3.8±1.2	4.6±1.5	3.8
	FeCAM (orig.)	7.2±2.0	7.6±1.3	8.4±2.8	6.2±0.8	5.6±0.3	7.0	3.2±1.2	3.4±0.9	4.2±1.4	3.7±1.4	4.1±1.3	3.7
Ours	**ProtoCIL**	**2.6**±1.2	**2.9**±1.0	**4.2**±1.8	**3.2**±0.8	6.3±2.5	**3.8**	**0.4**±0.1	**0.4**±0.1	3.9±1.0	**0.8**±0.2	3.3±0.7	**1.8**
						Plasticity (%) ↑							
Lower	Naive	79.1±0.7	75.4±0.8	54.4±1.6	76.3±0.4	69.7±0.3	71.0	98.4±0.7	97.5±0.7	53.6±5.2	95.8±1.2	94.8±0.8	88.0
Reg.	EWC	78.9±0.2	74.7±0.9	53.1±1.1	76.6±0.1	68.6±0.8	70.4	98.3±0.6	97.4±0.7	53.4±5.0	95.8±1.2	94.8±0.7	87.9
Replay	Replay	76.9±0.3	73.7±1.1	52.0±2.6	75.1±0.4	68.7±0.6	69.3	98.2±0.6	97.3±0.7	53.6±4.0	95.7±1.1	94.6±0.8	87.9
	iCaRL	76.3±0.6	73.6±2.2	53.4±2.5	74.5±0.7	68.1±0.8	69.2	**98.8**±0.8	**98.2**±0.9	54.8±6.8	**96.7**±1.4	**96.0**±1.4	88.9
	DER++	76.5±1.7	73.6±2.6	48.8±1.2	73.8±1.5	67.0±3.1	67.9	97.8±0.3	96.7±0.3	45.2±2.2	95.0±0.4	94.0±1.4	85.7
	RER	77.3±0.2	74.3±0.7	50.4±2.6	75.0±0.2	68.8±0.9	69.2	98.2±0.6	97.3±0.7	53.1±4.0	95.7±1.1	94.6±0.8	87.8
Subsp.	EASE	64.2±1.4	58.4±2.1	40.3±2.4	62.9±0.6	51.8±1.4	55.5	66.9±0.8	64.2±0.4	49.6±0.7	64.6±1.2	61.4±0.5	61.3
Proj.	RanPAC	80.1±1.8	75.7±1.5	27.6±3.1	76.8±1.4	70.5±1.9	66.1	88.5±1.1	86.3±1.1	28.9±1.2	89.1±0.8	84.8±1.1	75.5
Proto.	NCM	22.6±1.0	18.1±0.9	27.0±1.8	33.1±1.9	13.4±0.6	22.8	53.1±0.9	51.6±0.8	40.9±1.3	49.6±0.9	26.2±1.1	44.3
	FeCAM (ridge)	80.0±1.4	73.9±0.8	51.8±2.2	77.2±0.5	74.8±1.0	71.5	58.7±0.8	54.6±1.1	41.0±1.2	50.9±1.0	34.1±1.3	47.9
	FeCAM (orig.)	68.0±1.7	56.1±1.2	42.8±2.4	65.4±0.7	43.2±0.3	55.1	85.2±1.0	80.8±0.7	64.6±1.2	81.1±1.2	70.6±1.1	76.4
Ours	**ProtoCIL**	**83.5**±1.0	**78.9**±0.9	**66.2**±1.5	**81.0**±0.6	**77.0**±2.1	**77.3**	96.7±0.5	96.2±0.3	**86.2**±0.9	95.7±0.4	91.4±1.6	**93.2**

5 EXPERIMENTS

Overall Results. Table 1 summarizes the main results. ProtoCIL achieves the highest average accuracy on every encoder–dataset combination, reaching 74.3% on Enzyme Reaction and 91.8% on EC Number, 8.8% and 18.5% above the strongest baselines. It also attains the lowest average forgetting of 3.8% and 1.8% and the highest average plasticity of 77.3% and 93.2%. On EC Number, replay-based methods forget over 40%, while other low-forgetting methods reach at most 73.3% accuracy. NCM confirms that frozen prototype classification is stable, forgetting only 4.8% and 3.1%, but reaches just 18.7% and 41.6% accuracy, more than 50% below ProtoCIL. Even with per-encoder tuning, the strongest baselines reach only 71.3% and 76.6%, below ProtoCIL without tuning, as

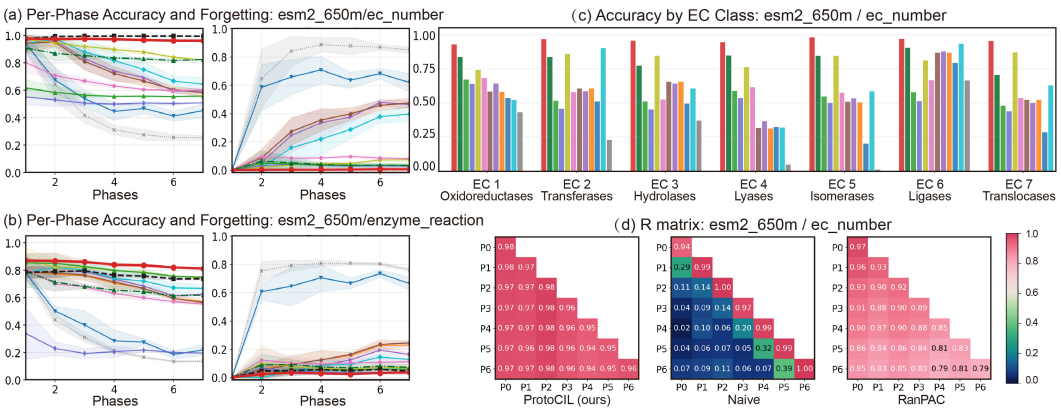


Figure 3: Results with ESM2-650M. (a, b) Average accuracy and forgetting over phases on EC Number and Enzyme Reaction. (c) Final accuracy by EC class on EC Number. (d) Phase-wise accuracy matrices on EC Number.

Table 2: Efficiency benchmark on EC Number with ESM2-650M at seed 0.

Method	Acc ↑	Fgt ↓	BP	Buffer	Storage/MB	Time/s
Naive	45.8	63.2	✓	0	23.7	153.5
EWC	58.4	48.4	✓	0	355.9	373.0
Replay	56.4	50.7	✓	2000	26.2	125.1
iCaRL	63.3	42.6	✓	2000	57.2	232.9
DER++	70.4	34.3	✓	2000	35.3	157.0
RER	56.8	50.4	✓	2000	26.2	358.3
EASE	65.5	9.0	✓	0	40.9	174.2
RanPAC	82.3	5.7	✗	0	921.3	5.8
NCM	50.8	1.5	✗	0	23.4	3.1
FeCAM (ridge)	55.8	2.1	✗	0	29.6	666.8
FeCAM (orig.)	82.4	1.8	✗	0	29.6	3.1
ProtoCIL	95.7	0.6	✗	0	260.9	467.9

Table 3: Sensitivity to the variance threshold τ in the r_τ computation. We report average accuracy (%) as mean ± standard deviation over three seeds.

	Enzyme Reaction			EC Number		
Encoder	τ=90	τ=95	τ=99	τ=90	τ=95	τ=99
ESM2 650M	81.3±0.5	81.6±0.7	**81.8**±0.8	96.3±0.5	**96.4**±0.6	96.1±0.5
ESM2 150M	**78.2**±0.3	76.8±0.8	77.7±0.8	**96.1**±0.2	96.0±0.2	95.7±0.1
SaProt	77.5±1.4	78.4±0.2	**78.7**±0.2	**95.1**±0.1	**95.1**±0.3	95.0±0.4
ProtBert	71.7±1.5	71.8±1.4	**72.4**±0.2	**89.2**±2.5	88.8±2.0	89.0±2.2
GearNet	**62.7**±0.0	**62.7**±0.0	**62.7**±0.0	78.1±6.7	**82.9**±0.0	77.5±7.6
Avg	74.3	74.3	***74.7***	91.0	***91.8***	90.7

shown in Table 16 of Appendix D. Figure 3 shows the phase-wise accuracy, forgetting, and stability, with results for all encoders in Figures 4–6 of Appendix D.

Computational efficiency. Table 2 compares efficiency on EC Number with the ESM2-650M encoder. ProtoCIL reaches 95.7% accuracy with 0.6% forgetting, requiring no backpropagation or replay buffer. Its storage of 260.9 MB is higher due to per-group covariance matrices, but it avoids trainable model expansion. Among baselines, FeCAM (orig.) and RanPAC reach the next-best accuracy of 82.4% and 82.3% in 3.1 s and 5.8 s, although RanPAC requires 921.3 MB for its Gram matrix. The backpropagation-free prototype baselines, NCM and both FeCAM variants, also forget little, but ProtoCIL is 13.3% more accurate than the best of them. ProtoCIL's runtime of 467.9 s is dominated by covariance updates over 38 groups.

Sensitivity to variance threshold. Table 3 reports the effect of varying the variance threshold τ used to compute the intrinsic rank r_τ in the group budget formula. All non-GearNet encoders show less than 1.5% variation across thresholds on both datasets. GearNet remains at G=1 across all thresholds on Enzyme Reaction, producing identical results. On EC Number, GearNet shows larger sensitivity under high-dimensional covariance estimation. We adopt τ=95 as it gives the best overall balance across both datasets. Likewise, as shown in Table 15 in the appendix, replacing the shrinkage threshold $\alpha < 1/2$ in the group budget with the stricter $\alpha < 1/3$ or $\alpha < 1/4$ changes average accuracy by at most 1.0%, indicating that $1/2$ is not a sensitive choice.

Module Ablation Analysis. Table 4 ablates ProtoCIL components across five encoders and two datasets. The Global pooled row uses a single covariance shared by all classes under the same shrinkage and classifier, serving as the global pooled covariance baseline. Adaptive grouping

Table 4: Ablation of ProtoCIL components. Avg Acc (%) is mean ± std over 3 seeds. G, S, and L denote adaptive grouping, Ledoit–Wolf shrinkage, and log-det normalization. Global pooled uses a single covariance shared by all classes (G=1), serving as the global pooled covariance baseline; G=C uses class-wise diagonal covariance.

				Enzyme Reaction (348 classes)						EC Number (4788 classes)					
Config	G	S	L	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg
Full	✓	✓	✓	**81.6**±0.7	**76.8**±0.8	**62.7**±0.0	**78.4**±0.2	71.8±1.4	**74.3**	**96.4**±0.6	**96.0**±0.2	82.9±0.0	**95.1**±0.3	**88.8**±2.0	**91.8**
Global pooled	✗	✓	✓	78.0±0.0	70.5±0.0	**62.7**±0.0	75.6±0.0	**73.7**±0.0	72.1	84.4±0.0	80.7±0.0	82.9±0.0	87.0±0.0	83.1±0.0	83.6
w/o Shrink	✓	✗	✓	69.8±0.7	66.4±1.1	32.3±1.9	70.1±1.1	58.2±1.3	59.4	91.3±0.4	90.7±0.3	48.3±1.1	86.6±0.9	80.9±0.5	79.6
w/o LogDet	✓	✓	✗	78.7±1.7	73.3±1.2	61.9±0.5	77.2±1.0	67.8±5.7	71.8	95.3±0.0	95.4±0.2	**86.6**±2.7	92.6±0.5	86.9±2.0	91.4
G=C	✗	✗	✗	22.5±0.0	17.2±0.0	17.3±0.0	37.2±0.0	13.6±0.0	21.6	55.5±0.0	50.4±0.0	22.2±0.0	47.8±0.0	37.1±0.0	42.6

Table 5: Protocol sanity check under alternative phase constructions

	Enzyme Reaction			EC Number		
Protocol	Acc ↑	Fgt ↓	Plast ↑	Acc ↑	Fgt ↓	Plast ↑
Random classes	72.0±8.0	4.6±0.7	75.9±8.2	91.3±6.2	0.8±0.7	91.8±5.6
Finer-EC buckets	72.0±7.3	4.8±0.5	75.9±7.2	91.8±5.9	1.5±1.5	92.9±4.6
Mixed hierarchy	70.3±8.8	5.1±1.1	74.5±7.9	91.7±5.9	0.9±0.7	92.3±5.3
Temporal	69.9±5.8	3.3±0.6	72.4±6.3	93.1±4.0	0.4±0.2	93.4±3.9
EC-first (main)	74.3±7.4	3.8±1.4	77.3±6.9	91.8±5.8	1.8±1.7	93.2±4.4

Table 6: Grouping strategy ablation averaged over five frozen encoders

	Enzyme Reaction		EC Number	
Strategy	Acc ↑	Fgt ↓	Acc ↑	Fgt ↓
EC hierarchy	72.1±5.9	5.2±1.2	83.6±2.3	4.6±0.5
K-means	69.3±14.5	6.8±6.1	86.7±18.4	4.3±8.5
Random	69.5±16.2	8.0±7.9	90.2±12.3	2.7±4.9
Validated	**74.3**±7.4	**3.8**±1.4	**91.8**±5.8	**1.8**±1.7

improves over it by 2.2% on Enzyme Reaction and 8.2% on EC Number. Shrinkage is the most critical component, as removing it reduces average accuracy by 14.9% and 12.2%, and GearNet accuracy by over 30%. Log-det normalization gives smaller gains of 2.5% and 0.4%. ProtoCIL also adapts to encoder geometry, as shown in Figure 7 of Appendix D. For GearNet with d=3072, the budget keeps a single group, avoiding unreliable multi-group estimation under limited samples, while for the other encoders grouping improves accuracy in 7 settings without manual tuning.

Robustness to Phase Construction. Table 5 evaluates ProtoCIL under alternative phase construction protocols beyond the main EC-first-digit setting. The random-class protocol assigns classes to phases while preserving phase sizes, the finer-EC protocol builds phases from finer EC-prefix buckets, and the mixed-hierarchy protocol mixes classes from different top-level EC groups within each phase. We further construct phases chronologically by the year each EC number was introduced (McDonald et al., 2009), simulating the real growth of the enzyme label space. ProtoCIL shows nearly unchanged performance on EC Number and remains strong on Enzyme Reaction, indicating robustness across phase constructions. Under the temporal protocol, Table 12 shows that ProtoCIL outperforms the strongest baseline by 7.7% and 11.0%. Per-encoder results and baseline comparisons under the other protocols are reported in Tables 13 and 11 of Appendix D.

Grouping Strategy Analysis. Table 6 compares four grouping strategies under the same group-count budget. EC hierarchy groups classes by enzyme taxonomy, K-means clusters class prototypes, and Random serves as a non-structural control. Validated grouping, used by ProtoCIL, accepts a split only when it improves held-out accuracy. It performs best on both datasets, suggesting that covariance groups are better selected by validation than by fixed hierarchy or unsupervised clustering. Per-encoder results are reported in Table 14 of Appendix D.

6 Conclusion

We introduce ProtoCIL, a class-incremental framework for protein function prediction over frozen pretrained representations. Our benchmark shows that vision-oriented CIL methods struggle across heterogeneous protein encoders and evolving enzyme label spaces. ProtoCIL addresses this gap with prototype classification and adaptive group-wise covariance estimation, balancing shared statistical stability with class-specific geometry. Across two datasets and five encoders, ProtoCIL improves retained accuracy and reduces forgetting without backpropagation or per-encoder tuning.

AI USE STATEMENT

Generative AI tools were used during manuscript preparation for language editing, LaTeX formatting, and assistance with code generation, review, and debugging. All AI-assisted text and code were reviewed by the authors, and all reported experimental results were independently verified against the executed code and saved outputs. Generative AI was not used to generate experimental data or to determine reported numerical results. The authors take full responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses previously published and publicly available protein function datasets and does not involve new human-subject recruitment, animal experiments, or biological sample collection. Data-release practices associated with the original datasets are described in their respective source publications. This study introduces a computational prediction framework and does not generate new biological interventions or experimental samples.

REPRODUCIBILITY STATEMENT

We provide the methodological derivations, detailed dataset construction, preprocessing, evaluation metrics, baseline configurations, implementation details, and additional robustness analyses in the appendix. All experiments use fixed random seeds and are reported over three seeds. The datasets used in this study are publicly available through their original publications. Code and scripts required to reproduce the experiments are provided in the anonymous supplementary material.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 556–572, 2018. doi: 10.1007/978-3-030-01252-6_33. URL https://doi.org/10.1007/978-3-030-01252-6_33.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero. Shrinkage algorithms for mmse covariance estimation. *Trans. Sig. Proc.*, 58(10):5016–5029, October 2010. ISSN 1053-587X. doi: 10.1109/TSP.2010.2053029. URL https://doi.org/10.1109/TSP.2010.2053029.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 01 2023. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL https://doi.org/10.1093/nar/gkac1052.

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

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Philip Hartout, Dexiong Chen, Paolo Pellizzoni, Carlos Oliver, and Karsten Borgwardt. Endowing protein language models with structural knowledge. *Bioinformatics*, 41(11):btaf582, 11 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf582. URL https://doi.org/10.1093/bioinformatics/btaf582.

Pedro Hermosilla, Marco Schäfer, Matej Lang, Gloria Fackelmann, Pere-Pau Vázquez, Barbora Kozlikova, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=l0mSUROpwY.

Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2 edition, 2002. doi: 10.1007/b98835. URL https://doi.org/10.1007/b98835.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. ISSN 0047-259X. doi: https://doi.org/10.1016/S0047-259X(03)00096-4. URL https://www.sciencedirect.com/science/article/pii/S0047259X03000964.

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

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2023. doi: 10.1109/TPAMI.2022.3213473.

Andrew G. McDonald, Sinéad Boyce, and Keith F. Tipton. Explorenz: the primary source of the iubmb enzyme list. *Nucleic Acids Research*, 37:D593–D597, 2009.

Mark D. McDonnell, Dong Gong, Amin Parveneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: random projections and pre-trained models for continual learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, 2013.

Darshan Patil, Pranshu Malviya, Mathieu Reymond, Quentin Fournier, and Sarath Chandar. Copep: Benchmarking continual pretraining for protein language models, 2026. URL https://arxiv.org/abs/2603.00253.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with tape. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017. doi: 10.1109/CVPR.2017.587.

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):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2016239118.

Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15:7407, 2024. doi: 10.1038/s41467-024-51844-2. URL https://doi.org/10.1038/s41467-024-51844-2.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual Decomposed Attention-Based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11909–11919, 2023.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 6987–7009, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1c42513b8895ab11fbbb5b7e8e6b6b02-Paper-Conference.pdf.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. MOS: Model surgery for pre-trained model-based class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20699–20707, 2025. doi: 10.1609/aaai.v39i19.34281. URL https://doi.org/10.1609/aaai.v39i19.34281.

Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L M Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2):243–246, 2024.

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

Quanziang Wang, Renzhen Wang, Yuexiang Li, Dong Wei, Hong Wang, Kai Ma, Yefeng Zheng, and Deyu Meng. Relational experience replay: Continual learning by adaptively tuning task-wise relationship. *IEEE Transactions on Multimedia*, 26:9683–9698, 2024. doi: 10.1109/TMM.2024.3397048. URL https://doi.org/10.1109/TMM.2024.3397048.

Wenkang Wang, Yunyan Shuai, Min Zeng, et al. Dpfunc: Accurately predicting protein function via deep learning with domain-guided structure information. *Nature Communications*, 16 (1):70, 2025b. doi: 10.1038/s41467-024-54816-8. URL https://doi.org/10.1038/s41467-024-54816-8.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the European Conference on Computer Vision*, pp. 631–648, 2022.

Edwin C. Webb. *Enzyme Nomenclature 1992: Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Academic Press, 1992.

Yuxin Yang, Abby Jerger, Song Feng, Zixu Wang, Christina Brasfield, Margaret S. Cheung, Jeremy Zucker, and Qiang Guan. Improved enzyme functional annotation prediction using contrastive learning with structural inference. *Communications Biology*, 7(1):1690, 2024. doi: 10.1038/s42003-024-07359-z. URL https://doi.org/10.1038/s42003-024-07359-z.

Wayland Yeung, Zhongliang Zhou, Liju Mathew, Nathan Gravel, Rahil Taujale, Brady O'Boyle, Mariah Salcedo, Aarya Venkat, William Lanzilotta, Sheng Li, and Natarajan Kannan. Tree visualizations of protein sequence embedding space enable improved functional clustering of diverse protein superfamilies. *Briefings in Bioinformatics*, 24(1):bbac619, 01 2023. ISSN 1477-4054. doi: 10.1093/bib/bbac619. URL https://doi.org/10.1093/bib/bbac619.

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. doi: 10.1126/science.adf2465. URL https://www.science.org/doi/abs/10.1126/science.adf2465.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19091–19101, 2023a. doi: 10.1109/ICCV51070.2023.01754.

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*, 2023b.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *Int. J. Comput. Vision*, 133(3):1012–1032, August 2024a. ISSN 0920-5691. doi: 10.1007/s11263-024-02218-0. URL https://doi.org/10.1007/s11263-024-02218-0.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 8363–8371, 2024b.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23554–23564, 2024c. doi: 10.1109/CVPR52733.2024.02223.

APPENDIX A: ADDITIONAL RELATED WORK

Protein representation learning and function prediction. Protein representation learning has moved from task-specific supervised models toward general-purpose pretrained encoders for downstream protein prediction. TAPE introduced a standardized benchmark for evaluating protein transfer learning across sequence-level and residue-level tasks (Rao et al., 2019). Rives et al. showed that large-scale unsupervised protein language models can encode structural and functional signals from protein sequences alone (Rives et al., 2021), while ESM2 further scaled protein language modeling and enabled accurate structure-informed prediction from sequence representations (Lin et al., 2023). ProtTrans similarly demonstrated that transformer-based language models trained on large protein corpora provide transferable representations for diverse biological tasks (Elnaggar et al., 2022). Beyond sequence-only encoders, GearNet learns protein representations via geometric structure pretraining on residue-level graphs (Zhang et al., 2023b), SaProt incorporates structure information through a structure-aware vocabulary (Su et al., 2024), and PST injects structural knowledge into protein language models (Hartout et al., 2025).

Despite these advances, pretrained protein representations are typically evaluated under static annotation settings, where downstream models assume a fixed label space and do not account for distribution shifts induced by evolving protein functional annotations.

Protein annotation learning under evolving label space. Protein function annotation can be viewed as prediction over large and evolving functional label spaces such as EC numbers. Prior work has advanced static protein annotation from several directions, including contrastive annotation learning (Yu et al., 2023), structure-based prediction from 3D protein conformations (Hermosilla et al., 2021), and models incorporating contact maps, domain-guided priors, or fine-tuned protein language models (Yang et al., 2024; Wang et al., 2025b; Schmirler et al., 2024). These studies improve protein annotation under fixed-label settings, but they do not explicitly address the sequential emergence of new functional categories or its effect on representation geometry and decision boundaries.

Class-incremental learning with pretrained encoders and prototype methods. Class-incremental learning aims to learn new classes sequentially while preserving performance on previously learned ones (Masana et al., 2023). Classical methods mitigate forgetting via regularization, replay, or exemplar-based learning, including EWC (Kirkpatrick et al., 2017), iCaRL (Rebuffi et al., 2017), A-GEM (Chaudhry et al., 2019), and DER++ (Buzzega et al., 2020). These approaches are effective in standard settings but often rely on trainable backbones or exemplar memory.

Recent work extends CIL to pretrained representations, where the main challenge shifts from representation learning to adaptation over frozen features. Prompt-based methods such as DualPrompt and CODA-Prompt adapt pretrained vision models via task-specific prompts (Wang et al., 2022; Smith et al., 2023), while SLCA, RanPAC, EASE, and MOS explore classifier alignment, random projection, and ensemble-based adaptation strategies (Zhang et al., 2023a; McDonnell et al., 2023; Zhou et al., 2024c; Sun et al., 2025). These results indicate that pretrained representations fundamentally reshape the design space of incremental learning.

Prototype-based classifiers provide a simple mechanism for incremental learning by representing each class with feature statistics, enabling new classes to be added without retraining the encoder (Mensink et al., 2013). However, in high-dimensional protein embedding spaces, reliable estimation of class statistics is challenging: class-wise covariance estimates are unstable under limited samples, while global covariance fails to capture heterogeneity across functional classes. Shrinkage estimators such as Ledoit–Wolf and OAS improve robustness in high-dimensional regimes (Ledoit & Wolf, 2004; Chen et al., 2010), and PCA-based intrinsic dimensionality provides a principled way to characterize effective feature rank (Jolliffe, 2002).

Existing work is largely divided between static protein annotation learning and class-incremental learning in vision-based pretrained settings. This leaves open the problem of class-incremental learning over frozen protein representations under evolving functional label spaces and heterogeneous embedding geometry.

APPENDIX B: BENCHMARK DETAILS

B.1 DATASETS AND PREPROCESSING

Table 7 summarizes the two datasets and preprocessing steps used in our benchmark. Both datasets are converted to a unified format where each sample contains an amino acid sequence, a functional label, and, when available, a path to a protein structure file. For Enzyme Reaction, multi-label EC vectors are converted to single labels via argmax. For EC Number, the first annotation is retained for multiply annotated proteins. AlphaFold v4 predicted structures are downloaded for EC Number, while Enzyme Reaction structures are distributed with the TorchDrug download. Samples without matched structures are excluded from structure-dependent encoders (GearNet-Edge and SaProt) but retained for sequence-only encoders. All PDB files are processed with Foldseek (van Kempen et al., 2024) to generate per-residue 3Di structural tokens required by SaProt.

Table 7: Dataset statistics and preprocessing. Counts report the final cached sequence-encoder splits used in the benchmark; structure-dependent encoders may exclude samples without matched structures.

Dataset	Classes	Train	Test	Train/class	Phases	Preprocessing
Enzyme Reaction	348	15,128	1,855	$\sim$43	7	Multi-label EC vectors converted to single labels via argmax
EC Number	4,788	73,259	9,112	$\sim$15	7	First EC annotation retained; AlphaFold v4 structures downloaded

B.2 ENCODERS AND EMBEDDING EXTRACTION

Table 8 lists the five frozen encoders and their extraction details. Each encoder is applied once per protein, and the resulting embeddings are cached and reused across all CIL methods. ESM2 models use mean pooling over non-special token representations, while ProtBert and SaProt use attention-weighted mean pooling. SaProt (Su et al., 2024) pairs each amino acid with its Foldseek 3Di structural token; proteins without matched 3Di sequences are excluded. GearNet-Edge (Zhang et al., 2023b) constructs multi-relational residue graphs from 3D coordinates and applies L2 normalization to prevent magnitude differences from dominating distance computations. Sequences exceeding encoder-specific length limits are truncated.

Table 8: Frozen encoders and embedding extraction details.

Encoder	Type	Input	Dim.	Pooling
ESM2-650M (Lin et al., 2023)	Sequence PLM	AA sequence (max 1022 residues)	1280	Mean pool, final layer
ESM2-150M (Lin et al., 2023)	Sequence PLM	AA sequence	640	Mean pool, final layer
ProtBert (Elnaggar et al., 2022)	Sequence PLM	Space-separated AA (max 1024 tokens)	1024	Attention-weighted mean
SaProt (Su et al., 2024)	Seq.+Struct.	AA + 3Di tokens (max 350 residues)	1280	Attention-weighted mean
GearNet-Edge (Zhang et al., 2023b)	Structure GNN	Residue graph from 3D coords	3072	Node mean; L2 norm.

B.3 CIL PROTOCOL

For both datasets, our main protocol partitions classes into seven phases by EC first digit, where each top-level category (oxidoreductases, transferases, hydrolases, lyases, isomerases, ligases, translocases) constitutes one phase. This serves as the main biologically structured phase construction. To test robustness beyond this setting, we also evaluate four alternative phase construction protocols. The random-class protocol randomly assigns classes to phases while preserving the phase sizes of the

Table 9: Baseline CIL methods and configurations.

Method	Strategy	Core Mechanism	Key Hyperparameters
Naive	Lower bound	Train on new phase only	Hidden=1024, lr=10^{-3}, 100 epochs
Joint MLP	Joint reference	Retrain on all data	Same as Naive
EWC (Kirkpatrick et al., 2017)	Regularization	Fisher-weighted L2 penalty	λ=2000
Replay (Chaudhry et al., 2019)	Replay	Fixed-size reservoir buffer	Buffer=2,000
iCaRL (Rebuffi et al., 2017)	Replay + Distillation	Exemplar replay + distillation	Buffer=2,000
DER++ (Buzzega et al., 2020)	Replay + Distillation	Logit replay + label replay	Buffer=2,000, α=0.5, β=0.5
RER (Wang et al., 2024)	Adaptive replay	Gradient-cosine adaptive α	Buffer=2,000
RanPAC (McDonnell et al., 2023)	Projection	Random projection + ridge regression	D=12800, λ=1.0
EASE (Zhou et al., 2024c)	Subspace	Learned per-task adapter + prototypes	Rank=256, lr=10^{-3}, 100 epochs
NCM (Mensink et al., 2013)	Prototype	Euclidean nearest class mean	None
FeCAM (ridge) (Goswami et al., 2023)	Prototype	Shared Mahalanobis distance	Ridge λ=10^{-4}
FeCAM (orig.) (Goswami et al., 2023)	Prototype	Tukey transform + shrinkage + normalization	γ_1=γ_2=1

main protocol. The finer-EC protocol first groups classes by finer EC-prefix buckets and then assembles phases from these buckets. The mixed-hierarchy protocol constructs phases by mixing classes from different top-level EC groups, reducing the alignment between phase boundaries and enzyme taxonomy. The temporal protocol orders classes by the year their EC number was created in the IUBMB enzyme list (McDonald et al., 2009) and splits them chronologically into phases of the same sizes. Classes with provisional EC numbers that lack a creation year are excluded.

For each protocol except the temporal one, phase order is shuffled per random seed using a deterministic permutation, and all methods within the same seed observe the same order. At each phase, MLP-based methods optimize a fixed global classifier head on the new-phase embeddings, with replay buffer access where applicable, while prototype-based and projection-based methods compute class statistics or solve closed-form classifiers directly over the observed classes. After each phase, accuracy is evaluated on test samples from all observed phases. All results are averaged over three random seeds (0, 7, 42) with standard deviations reported.

B.4 Baselines and Hyperparameters

Table 9 lists the baseline methods and configurations. All MLP-based methods use a two-layer classifier with hidden dimension 1024, trained with Adam (learning rate 10^{-3}) for 100 epochs per phase. Replay, iCaRL, DER++, and RER use a 2,000-sample reservoir buffer. DER++ uses α=0.5 and β=0.5, and EWC uses a Fisher penalty weight of λ=2000. FeCAM (ridge) uses a global shared covariance with ridge regularization (λ=10^{-4}), and FeCAM (orig.) follows the original configuration with γ_1=γ_2=1. RanPAC uses a fixed Gaussian random projection with D=12800 and ridge parameter λ=1.0. EASE trains a low-rank adapter (rank 256) per phase with the same learning rate and epoch count as MLP-based methods. All hyperparameters are fixed across every encoder-dataset configuration, and results with per-encoder tuning are reported in Appendix D.

B.5 Sequence Similarity Across Phases

To assess whether proteins from different phases are closely related, we perform an all-versus-all sequence search over all proteins of each dataset with MMseqs2 (Steinegger & Söding, 2017) (E-value 10^{-3}) and record, for each protein, its maximum sequence identity to any other protein in a different phase and in the same phase. Table 10 shows that high-identity pairs almost exclusively occur within the same phase, so each phase introduces proteins that are largely distinct in sequence from those of other phases.

Table 10: Fraction of proteins (%) whose maximum sequence identity to a protein from a different phase or from the same phase exceeds a given threshold.

	Enzyme Reaction		**EC Number**	
Identity	Other phases	Same phase	Other phases	Same phase
$\geq 30\%$	14.0	86.0	15.0	99.8
$\geq 50\%$	2.9	57.4	1.4	86.5
$\geq 70\%$	1.3	36.6	0.0	24.0
$\geq 90\%$	0.8	21.0	0.0	22.1

APPENDIX C: DERIVATION OF THE GROUP COUNT FORMULA

We present a self-contained derivation of the principled grouping formula used in ProtoCIL. The derivation proceeds in five stages: Stage 1 establishes a diagnostic for covariance quality; Stage 2 expands the shrinkage intensity as a function of sample size; Stage 3 converts the diagnostic into a sample requirement; Stage 4 solves for the number of groups G; Stage 5 addresses rank-deficiency and adaptive splitting.

C.1 STAGE 1: A DIAGNOSTIC FOR COVARIANCE QUALITY

Setup. ProtoCIL partitions C classes into G groups and estimates a pooled covariance matrix $\hat{\mathbf{\Sigma}}_g \in \mathbb{R}^{d\times d}$ per group g. Each group contains N_g samples drawn from its constituent classes. To classify via the Mahalanobis distance, we need $\hat{\mathbf{\Sigma}}_g$ to faithfully reflect the true within-group covariance $\mathbf{\Sigma}_g$. The question is: *how large must N_g be for this estimate to be reliable?*

The Ledoit–Wolf estimator. When N_g is not much larger than d, the sample covariance $\hat{\mathbf{S}} = \frac{1}{N_g-1}\sum_{i=1}^{N_g}(\mathbf{x}_i-\bar{\mathbf{x}})(\mathbf{x}_i-\bar{\mathbf{x}})^\top$ is noisy or singular. (We write $\hat{\mathbf{S}}$ for the normalized sample covariance; this corresponds to $\mathbf{S}/(N_g-1)$ where $\mathbf{S}$ is the scatter matrix in the main text.) The Ledoit–Wolf (LW) shrinkage estimator (Ledoit & Wolf, 2004) regularizes it by mixing with an isotropic target:

$$\hat{\mathbf{\Sigma}} \;=\; (1-\alpha)\,\hat{\mathbf{S}} \;+\; \alpha\,\frac{\operatorname{tr}(\hat{\mathbf{S}})}{d}\,\mathbf{I}, \tag{C1}$$

where the shrinkage intensity $\alpha \in [0,1]$ is determined analytically from the data. Intuitively, α interpolates between two extremes:

- $\alpha \to 0$: the estimate reduces to the sample covariance $\hat{\mathbf{S}}$, preserving all directional information but suffering from estimation noise.
- $\alpha \to 1$: the estimate collapses to a scaled identity $\frac{\operatorname{tr}(\hat{\mathbf{S}})}{d}\mathbf{I}$, which is maximally stable but erases all directional structure.

Choosing the threshold. For the Mahalanobis classifier, directional structure is essential: the covariance must encode which directions carry large vs. small variance. If $\alpha \geq 1/2$, the isotropic target dominates and the sample covariance contributes at most half the weight, meaning directional information is no longer the primary signal. We therefore require:

$$\alpha \;<\; \tfrac{1}{2}. \tag{C2}$$

This is a natural midpoint criterion: the sample covariance must carry *strictly more* weight than the isotropic regularizer. The choice of $1/2$ as the threshold is the unique point at which neither component dominates, and any stricter threshold (e.g., $\alpha < 1/3$) would impose unnecessarily large sample requirements without qualitative benefit.

C.2 STAGE 2: EXPANDING α AS A FUNCTION OF N_g

The OAS formula. To convert the condition equation C2 into a concrete requirement on N_g, we need an explicit expression for α in terms of N_g and properties of the true covariance $\mathbf{\Sigma}$. The Oracle Approximating Shrinkage (OAS) estimator (Chen et al., 2010) provides:

$$\alpha^* \;=\; \frac{\left(1-\frac{2}{d}\right)\operatorname{tr}(\mathbf{\Sigma}^2)+\operatorname{tr}^2(\mathbf{\Sigma})}{\left(N_g+1-\frac{2}{d}\right)\left(\operatorname{tr}(\mathbf{\Sigma}^2)-\frac{\operatorname{tr}^2(\mathbf{\Sigma})}{d}\right)}. \tag{C3}$$

Spectral structure of protein embeddings. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$ be the eigenvalues of $\mathbf{\Sigma}$. Protein embeddings produced by pre-trained encoders (ESM-2, GearNet, SaProt, etc.) are known to concentrate their variance on a low-dimensional subspace. That is, there exists an effective dimension

$d_{\text{eff}} \ll d$ such that $\lambda_i \approx 0$ for all $i > d_{\text{eff}}$. Under this structure:

$$\operatorname{tr}(\mathbf{\Sigma}^2) = \sum_{i=1}^{d} \lambda_i^2 \;\approx\; \sum_{i=1}^{d_{\text{eff}}} \lambda_i^2, \tag{C4a}$$

$$\operatorname{tr}(\mathbf{\Sigma}) = \sum_{i=1}^{d} \lambda_i \;\approx\; \sum_{i=1}^{d_{\text{eff}}} \lambda_i. \tag{C4b}$$

Uniform eigenvalue approximation. To obtain a closed-form relationship between α^* and N_g, we introduce a mild simplification: we approximate the d_{eff} signal eigenvalues as roughly equal, $\lambda_1 \approx \cdots \approx \lambda_{d_{\text{eff}}} \equiv \bar{\lambda}$. Under this assumption:

$$\operatorname{tr}(\mathbf{\Sigma}^2) \approx d_{\text{eff}}\, \bar{\lambda}^2, \tag{C5a}$$

$$\operatorname{tr}^2(\mathbf{\Sigma}) \approx \left(d_{\text{eff}}\, \bar{\lambda}\right)^2 = d_{\text{eff}}^2\, \bar{\lambda}^2. \tag{C5b}$$

Observe the key relationship between these two quantities:

$$\operatorname{tr}^2(\mathbf{\Sigma}) \;\approx\; d_{\text{eff}} \cdot \operatorname{tr}(\mathbf{\Sigma}^2). \tag{C5c}$$

Substitution into the OAS formula. We now substitute equation C5a–equation C5c into equation C3.

Numerator of equation C3:

$$\begin{aligned}
&\left(1 - \tfrac{2}{d}\right) \operatorname{tr}(\mathbf{\Sigma}^2) + \operatorname{tr}^2(\mathbf{\Sigma}) \\
&\quad\approx\; \left(1 - \tfrac{2}{d}\right) \operatorname{tr}(\mathbf{\Sigma}^2) + d_{\text{eff}} \cdot \operatorname{tr}(\mathbf{\Sigma}^2) && \text{(by equation C5c)} \\
&\quad=\; \operatorname{tr}(\mathbf{\Sigma}^2) \left(1 - \tfrac{2}{d} + d_{\text{eff}}\right) \\
&\quad\approx\; \operatorname{tr}(\mathbf{\Sigma}^2) \left(1 + d_{\text{eff}}\right) && \text{(C6a)}
\end{aligned}$$

where the last step drops the $O(1/d)$ term since $d \gg 1$ (e.g., $d = 640$ for ESM-2-150M, $d = 1280$ for ESM-2-650M).

Denominator of equation C3:

$$\begin{aligned}
&\left(N_g + 1 - \tfrac{2}{d}\right) \left(\operatorname{tr}(\mathbf{\Sigma}^2) - \tfrac{\operatorname{tr}^2(\mathbf{\Sigma})}{d}\right) \\
&\quad\approx\; (N_g + 1) \left(\operatorname{tr}(\mathbf{\Sigma}^2) - \tfrac{d_{\text{eff}} \cdot \operatorname{tr}(\mathbf{\Sigma}^2)}{d}\right) && \text{(by equation C5c, dropping } \tfrac{2}{d}\text{)} \\
&\quad=\; (N_g + 1) \operatorname{tr}(\mathbf{\Sigma}^2) \left(1 - \tfrac{d_{\text{eff}}}{d}\right). && \text{(C6b)}
\end{aligned}$$

Since d_{eff} is smaller than d in our encoder settings, we use $(1 - d_{\text{eff}}/d) \approx 1$ as a simplifying approximation.

Combining numerator and denominator:

$$\begin{aligned}
\alpha^* \;&\approx\; \frac{\operatorname{tr}(\mathbf{\Sigma}^2)(1 + d_{\text{eff}})}{(N_g + 1) \operatorname{tr}(\mathbf{\Sigma}^2) \cdot 1} \\
&=\; \frac{1 + d_{\text{eff}}}{N_g + 1} \\
&\approx\; \frac{d_{\text{eff}}}{N_g} && \text{(C7)}
\end{aligned}$$

where the final step uses $d_{\text{eff}} \gg 1$ (empirically $d_{\text{eff}} \geq 9$) so that $1 + d_{\text{eff}} \approx d_{\text{eff}}$, and similarly $N_g + 1 \approx N_g$ for groups of meaningful size.

Interpretation. Equation equation C7 reveals a clean inverse relationship: the shrinkage intensity is approximately the ratio of effective dimensionality to sample size. This makes physical sense: more samples relative to the intrinsic dimension means less need for regularization.

C.3 Stage 3: Converting to a Sample Requirement

Substituting equation C7 into the diagnostic condition equation C2:

$$\frac{d_{\text{eff}}}{N_g} \;<\; \frac{1}{2}. \tag{C8}$$

Rearranging:

$$N_g \;>\; 2\,d_{\text{eff}}. \tag{C9}$$

This is our core sample requirement: each group must contain at least $2\,d_{\text{eff}}$ samples for its pooled covariance to be dominated by data rather than by the isotropic prior.

Measuring d_{eff} from data. The effective dimension d_{eff} is not a hyperparameter but an empirical quantity. We measure it as follows. Pool all within-class centered embeddings from all currently seen classes into a matrix $\mathbf{X}_{\text{centered}} \in \mathbb{R}^{N\times d}$ and compute its singular value decomposition, yielding singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$. Since the eigenvalues of the empirical covariance are $\lambda_i \propto \sigma_i^2$, we define the effective dimension via a 95% explained-variance threshold:

$$r_{95} \;=\; \min\left\{ r \;:\; \frac{\sum_{i=1}^{r}\sigma_i^2}{\sum_{j=1}^{d}\sigma_j^2} \;\geq\; 0.95 \right\}. \tag{C10}$$

The choice of 95% balances two concerns: a threshold too low (e.g., 80%) would undercount meaningful dimensions, producing overly many groups with noisy estimates; a threshold too high (e.g., 99%) would include noise dimensions, producing overly few groups and losing class-level resolution.

Setting $d_{\text{eff}} = r_{95}$ and substituting into equation C9:

$$N_g \;>\; 2\,r_{95}. \tag{C11}$$

C.4 Stage 4: Solving for the Target Group Count G_{target}

From samples per group to target group count. Suppose we partition C classes into G equal-sized groups. The total number of samples from all seen classes is $N_{\text{total}} = C\bar{n}$, where $\bar{n}$ is the mean number of samples per class. Each group then contains:

$$N_g \;=\; \frac{C\bar{n}}{G}. \tag{C12}$$

Substituting into equation C11:

$$\frac{C\bar{n}}{G} \;>\; 2\,r_{95}. \tag{C13}$$

Solving for G:

$$G \;<\; \frac{C\bar{n}}{2\,r_{95}}. \tag{C14}$$

We use this bound to define a rounded target group count, subject to the natural constraint $G \leq C$ (we cannot have more groups than classes):

$$\boxed{G_{\text{target}} \;=\; \min\left(C,\; \left\lfloor \frac{C\,\bar{n}}{2\,r_{95}} \right\rceil\right)} \tag{C15}$$

All quantities (C, $\bar{n}$, r_{95}) in the group-count formula are computed from data and updated after each phase as new classes arrive, requiring no per-encoder tuning.

The actual number of groups grows incrementally toward G_{target} via budget-guided validated splits, starting from a single group (G=1) at the base phase.

C.5 Stage 5: Rank-Deficiency Guard and Adaptive Splitting

The condition $\alpha < 1/2$ (Stage 1) ensures that the sample covariance dominates the isotropic prior. However, the LW estimator equation C1 faces a separate, more fundamental constraint from linear algebra that we now formalize.

Lemma .1 (Null-space degeneracy)**.** *If $N_g < d$, then the regularized covariance $\hat{\mathbf{\Sigma}}$ from equation C1 has at least $d - N_g + 1$ eigenvalues equal to $\alpha\,\mathrm{tr}(\hat{\mathbf{S}})/d$. These eigenvalues depend only on the global trace and the ambient dimension, not on which classes are assigned to the group. Consequently, the corresponding eigendirections carry no group-specific discriminative information.*

Proof. When $N_g < d$, the sample covariance has $\mathrm{rank}(\hat{\mathbf{S}}) \le N_g - 1 < d$, so its null space $\mathcal{N} = \{\mathbf{v} : \hat{\mathbf{S}}\,\mathbf{v} = \mathbf{0}\}$ has dimension at least $d - N_g + 1$. For any $\mathbf{v} \in \mathcal{N}$, substituting into equation C1:

$$\begin{aligned}\hat{\mathbf{\Sigma}}\,\mathbf{v} &= (1-\alpha)\,\underbrace{\hat{\mathbf{S}}\,\mathbf{v}}_{=\mathbf{0}} + \alpha\,\frac{\mathrm{tr}(\hat{\mathbf{S}})}{d}\,\mathbf{v} \\ &= \alpha\,\frac{\mathrm{tr}(\hat{\mathbf{S}})}{d}\,\mathbf{v}. \end{aligned} \tag{C16}$$

Thus every null-space vector is an eigenvector of $\hat{\mathbf{\Sigma}}$ with eigenvalue $\alpha\,\mathrm{tr}(\hat{\mathbf{S}})/d$. This eigenvalue is determined entirely by: (i) the shrinkage intensity α, which is a function of the spectral properties of the pooled data; (ii) $\mathrm{tr}(\hat{\mathbf{S}})$, the total variance of the pooled samples; (iii) the ambient dimension d.

These null-space directions are isotropic within each group and carry no group-specific directional structure beyond a scalar variance level. Therefore, they provide little discriminative benefit over a single shared covariance. □

Practical consequence. Lemma .1 imposes the hard constraint $N_g \ge d$, i.e., $C\bar{n}/G_{\mathrm{target}} \ge d$. If the G_{target} computed from equation C15 violates this constraint, we apply the following correction: repeatedly halve G_{target} (i.e., $G_{\mathrm{target}} \leftarrow \lfloor G_{\mathrm{target}}/2 \rfloor$) until $C\bar{n}/G_{\mathrm{target}} \ge d$, or set $G_{\mathrm{target}} = 1$ as a fallback. In practice, this correction is rarely triggered because $r_{95} \ll d$ implies the bound from equation C14 already produces a sufficiently small G_{target}.

Lemma .2 (Adaptive split threshold)**.** *During the validated split refinement, a group with k classes and approximately $k\bar{n}$ samples is bisected into two children. For both children to satisfy the covariance quality condition equation C11, the parent group must contain at least M_{split} classes, where:*

$$M_{\mathrm{split}} \;=\; 2\,\max\!\left(2,\; \left\lceil \frac{2\,r_{95}}{\bar{n}} \right\rceil\right). \tag{C17}$$

Proof. Suppose a parent group with k classes is bisected into two children, each receiving approximately $k/2$ classes. We derive the minimum parent size by reasoning about each child independently.

Step 1: Covariance quality per child. Each child has $k_{\mathrm{child}} \approx k/2$ classes and $k_{\mathrm{child}}\,\bar{n}$ samples. Applying the sample requirement equation C11 to a single child:

$$k_{\mathrm{child}}\,\bar{n} \;>\; 2\,r_{95} \quad\Longrightarrow\quad k_{\mathrm{child}} \;\ge\; \left\lceil \frac{2\,r_{95}}{\bar{n}} \right\rceil. \tag{C18}$$

Step 2: Minimum group size. A covariance estimate from a single class is degenerate (zero within-class scatter for $n = 1$, or rank-1 for small n). Each child must therefore contain at least 2 classes:

$$k_{\mathrm{child}} \;\ge\; \max\!\left(2,\; \left\lceil \frac{2\,r_{95}}{\bar{n}} \right\rceil\right). \tag{C18b}$$

Step 3: Parent constraint. Since the parent must accommodate two such children ($k = 2\,k_{\mathrm{child}}$):

$$M_{\mathrm{split}} \;=\; 2\,\max\!\left(2,\; \left\lceil \frac{2\,r_{95}}{\bar{n}} \right\rceil\right). \tag{C19}$$

□

Summary. The complete derivation chain is:

$$\underbrace{\alpha < \tfrac{1}{2}}_{\text{diagnostic}} \xrightarrow{\text{OAS}} \underbrace{\tfrac{d_{\text{eff}}}{N_g} < \tfrac{1}{2}}_{\text{expansion}} \implies \underbrace{N_g > 2\,r_{95}}_{\text{sample bound}} \implies \underbrace{G_{\text{target}} = \min\Big(C,\ \Big\lfloor \tfrac{C\bar{n}}{2\,r_{95}} \Big\rceil\Big)}_{\text{target group count}}$$

augmented by the rank-deficiency guard $N_g \geq d$ (Lemma .1) and the split threshold M_{split} (Lemma .2). The group-count quantities are computed from the currently seen data and updated after each phase, requiring no per-encoder tuning.

Algorithm 1 ProtoCIL Procedure

Require: Phases $\{P_t\}_{t=0}^{T-1}$, frozen encoder f_θ, held-out fraction η
Ensure: Prototypes $\{\boldsymbol{\mu}_c\}$, group mapping $g(\cdot)$, per-group $\boldsymbol{\Sigma}_g^{-1}$, $\log|\boldsymbol{\Sigma}_g|$

for each phase P_t **do**
  ▷ Prototype and scatter computation
  for each new class $c \in P_t$ **do**
    Reserve η fraction as held-out; compute $\boldsymbol{\mu}_c$ and $\mathbf{S}_c$ from remainder
  end for
  ▷ Incremental grouping
  if $t = 0$ **then**
    Place all classes into a single group g_0
  else
    Assign each new class to nearest group; update $\mathbf{S}_{g(c)} \leftarrow \mathbf{S}_{g(c)} + \mathbf{S}_c$
  end if
  ▷ Target group count and splitting
  Compute r_{95}, G_{target}, split budget B, and M_{split}
  while $B > 0$ and $\exists\, g$ with $|g| \geq M_{\text{split}}$ **do**
    Search top-k PC directions at quartile thresholds
    if held-out accuracy improves **then**
      Accept split; $B \leftarrow B - 1$
    end if
  end while
  ▷ Mahalanobis classification
  Finalise per-group $\boldsymbol{\Sigma}_g$ via LW shrinkage; classify via equation 11
end for

APPENDIX D: ADDITIONAL EXPERIMENTAL RESULTS

Table 11: Protocol sanity check with baselines under alternative phase constructions. We report encoder-average Avg Acc, Forgetting, and Plasticity (%, mean±std). ProtoCIL values are evaluated under the same random, finer-EC, and mixed phase-construction protocols as the baselines.

Dataset	Method	Random classes			Finer-EC buckets			Mixed hierarchy		
		Acc ↑	**Fgt ↓**	**Plast ↑**	**Acc ↑**	**Fgt ↓**	**Plast ↑**	**Acc ↑**	**Fgt ↓**	**Plast ↑**
Enzyme Reaction	FeCAM (ridge)	60.6±11.8	12.1±1.8	70.9±12.2	62.0±12.3	8.7±1.3	69.4±11.7	61.2±12.2	10.8±0.6	70.4±12.6
	DER++	55.3±10.5	19.9±5.7	72.0±6.0	58.4±11.1	16.5±5.1	71.7±7.2	57.0±9.9	19.6±5.1	73.5±5.8
	RanPAC	60.2±20.1	12.6±0.6	70.8±19.9	62.6±21.9	6.4±2.0	67.9±20.6	61.5±20.8	9.8±0.8	69.9±20.5
	ProtoCIL	**72.0**±8.0	**4.6**±0.7	**75.9**±8.2	**72.0**±7.3	**4.8**±0.5	**75.9**±7.2	**70.3**±8.8	**5.1**±1.1	**74.5**±7.9
EC Number	FeCAM (ridge)	44.7±10.2	6.5±0.8	50.3±10.8	49.5±10.3	3.8±0.4	52.7±10.4	46.2±10.5	5.4±0.8	50.9±11.0
	DER++	63.6±12.4	41.3±11.9	**98.5**±3.1	65.3±13.4	39.0±12.6	**98.5**±3.2	62.2±12.4	42.8±11.8	**98.0**±4.2
	RanPAC	68.7±26.0	9.6±2.4	76.9±27.5	69.3±27.8	7.1±1.7	75.4±28.9	68.9±26.1	11.1±2.5	78.5±27.6
	ProtoCIL	**91.3**±6.2	**0.8**±0.7	91.8±5.6	**91.8**±5.9	**1.5**±1.5	92.9±4.6	**91.7**±5.9	**0.9**±0.7	92.3±5.3

Table 12: Temporal protocol with baselines. Classes arrive in chronological order of EC number creation. We report encoder-average Avg Acc, Forgetting, and Plasticity (%, mean±std across encoders).

Method	Enzyme Reaction			EC Number		
	Acc ↑	Fgt ↓	Plast ↑	Acc ↑	Fgt ↓	Plast ↑
FeCAM (ridge)	62.2±9.3	6.2±0.8	67.5±9.4	57.8±8.1	3.0±0.8	60.4±8.2
FeCAM (orig.)	47.3±10.3	5.6±1.1	52.2±11.1	82.1±6.0	1.4±0.2	83.3±6.0
DER++	46.3±8.1	27.0±5.6	69.4±3.8	51.3±8.3	56.4±9.0	99.6±0.6
RanPAC	58.2±18.2	4.2±0.5	61.7±18.1	63.3±21.0	1.8±0.4	64.7±21.2
ProtoCIL	69.9±5.8	3.3±0.6	72.4±6.3	93.1±4.0	0.4±0.2	93.4±3.9

Table 13: Protocol sanity check by encoder. We report mean ± standard deviation over three seeds for Avg Acc, Forgetting, and Plasticity (%) under different phase construction protocols. Random, finer-EC, mixed, and temporal protocols test whether ProtoCIL's gains depend on the EC-first-digit phase construction; EC denotes the main phase construction used in the main benchmark.

	Enzyme Reaction (348 classes)						EC Number (4788 classes)					
Protocol	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg
					Average (%) ↑							
Random	80.0±0.7	75.7±1.9	58.8±1.0	73.7±1.6	**71.8**±2.8	72.0±8.0	96.1±0.1	95.8±0.1	82.0±0.1	94.8±0.5	87.7±1.2	91.3±6.2
Finer-EC	79.5±1.4	75.0±0.5	60.2±1.2	74.8±0.3	70.5±1.4	72.0±7.3	**96.6**±0.1	95.8±0.2	82.8±0.1	94.9±0.7	88.8±0.5	91.8±5.9
Mixed	77.5±2.5	74.9±2.5	55.3±4.3	73.4±1.5	70.3±0.9	70.3±8.8	96.5±0.3	95.7±0.2	83.0±0.3	95.1±0.5	88.3±0.0	91.7±5.9
Temporal	76.5±0.7	72.0±0.7	59.6±0.1	73.1±0.8	68.1±0.9	69.9±5.8	96.4±0.2	**96.1**±0.4	**85.9**±0.2	**95.7**±0.3	**91.5**±0.5	**93.1**±4.0
EC (main)	**81.6**±0.7	**76.8**±0.8	**62.7**±0.0	**78.4**±0.2	**71.8**±1.4	**74.3**±7.4	96.4±0.6	96.0±0.2	82.9±0.0	95.1±0.3	88.8±2.0	91.8±5.8
					Forgetting (%) ↓							
Random	4.4±1.4	4.3±2.0	3.8±0.1	4.8±0.9	5.7±0.5	4.6±0.7	**0.2**±0.1	0.3±0.1	1.9±0.1	0.5±0.2	1.1±0.3	0.8±0.7
Finer-EC	4.7±1.9	4.8±1.4	4.5±1.2	4.4±1.3	5.7±0.8	4.8±0.5	0.3±0.1	0.4±0.2	3.5±1.1	0.8±0.3	2.5±1.0	1.5±1.5
Mixed	4.2±1.2	4.9±0.5	7.0±4.8	4.4±0.6	5.2±0.5	5.1±1.1	0.3±0.1	0.3±0.0	1.9±0.1	0.6±0.1	1.3±0.4	0.9±0.7
Temporal	3.6±0.2	3.9±0.6	**2.4**±0.3	3.8±0.5	**2.7**±0.4	**3.3**±0.6	**0.2**±0.1	**0.2**±0.1	**0.8**±0.0	**0.3**±0.0	**0.6**±0.2	**0.4**±0.2
EC (main)	**2.6**±1.2	**2.9**±1.0	4.2±1.8	**3.2**±0.8	6.3±2.5	3.8±1.4	0.4±0.1	0.4±0.1	3.9±1.0	0.8±0.2	3.3±0.7	1.8±1.7
					Plasticity (%) ↑							
Random	**83.7**±1.4	**79.3**±0.3	62.0±1.0	77.8±1.6	76.6±2.5	75.9±8.2	96.1±0.1	96.0±0.2	83.6±0.1	95.1±0.4	88.2±1.4	91.8±5.6
Finer-EC	83.3±0.6	78.5±0.2	64.0±0.2	78.3±1.0	75.3±0.7	75.9±7.2	96.7±0.1	95.9±0.3	85.8±0.9	95.5±0.7	90.8±1.2	92.9±4.6
Mixed	81.1±2.4	79.0±2.0	61.1±1.9	76.8±2.1	74.5±0.6	74.5±7.9	**96.8**±0.2	95.8±0.2	84.6±0.2	95.3±0.8	89.0±0.6	92.3±5.3
Temporal	79.6±0.9	75.4±0.2	61.5±0.3	75.9±0.3	69.9±1.2	72.4±6.3	96.6±0.2	**96.2**±0.3	**86.6**±0.2	**95.9**±0.4	**91.5**±0.4	**93.4**±3.9
EC (main)	83.5±1.0	78.9±0.9	**66.2**±1.5	**81.0**±0.6	**77.0**±2.1	**77.3**±6.9	96.7±0.5	**96.2**±0.3	86.2±0.9	95.7±0.4	91.4±1.6	93.2±4.4

Table 14: Full grouping strategy ablation by encoder. We report mean ± standard deviation over three seeds for Avg Acc and Forgetting (%) under each grouping strategy across two datasets and five frozen encoders. The table expands the averaged main-text ablation and shows that validated grouping provides the most stable cross-encoder trade-off, especially avoiding the severe degradation of random or k-means grouping on the high-dimensional GearNet encoder. Bold indicates the unique best mean value within each encoder–dataset column. EC hierarchy keeps a single group in all runs and thus matches Global pooled in Table 4.

	Enzyme Reaction (348 classes)						EC Number (4788 classes)					
Grouping Strategy	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg
					Average (%) ↑							
EC hierarchy	78.0±0.0	70.5±0.0	62.7±0.0	75.6±0.0	73.7±0.0	72.1±5.9	84.4±0.0	80.7±0.0	82.9±0.0	87.0±0.0	83.1±0.0	83.6±2.3
K-means	79.7±2.4	**76.9**±0.6	43.9±3.0	73.8±0.6	72.2±1.3	69.3±14.5	96.3±0.1	94.9±0.3	53.9±6.0	94.9±0.3	93.6±0.4	86.7±18.4
Random	80.8±1.3	76.5±0.8	40.9±4.8	74.5±0.8	**74.9**±0.7	69.5±16.2	**96.8**±0.1	95.7±0.1	68.2±8.6	**95.8**±0.1	**94.5**±0.0	90.2±12.3
Validated	**81.6**±0.7	76.8±0.8	**62.7**±0.0	**78.4**±0.2	71.8±1.4	**74.3**±7.4	96.4±0.6	**96.0**±0.2	**82.9**±0.0	95.1±0.3	88.8±2.0	**91.8**±5.8
					Forgetting (%) ↓							
EC hierarchy	5.3±1.4	7.1±1.5	4.2±2.2	5.1±1.6	4.3±1.4	5.2±1.2	5.4±1.3	4.8±1.9	3.9±1.3	4.4±1.4	4.4±1.6	4.6±0.5
K-means	3.0±1.0	3.7±0.8	17.6±2.4	4.8±2.9	4.8±2.4	6.8±6.1	0.3±0.1	0.6±0.1	19.5±2.0	0.8±0.1	**0.6**±0.5	4.3±8.5
Random	3.3±1.9	4.4±2.4	22.0±2.9	5.8±3.0	4.3±1.5	8.0±7.9	**0.2**±0.1	0.7±0.3	11.5±3.1	**0.5**±0.1	0.8±0.2	2.7±4.9
Validated	**2.6**±1.2	**2.9**±1.0	**4.2**±1.8	**3.2**±0.8	6.3±2.5	**3.8**±1.4	0.4±0.1	**0.4**±0.1	**3.9**±1.0	0.8±0.2	3.3±0.7	**1.8**±1.7

Table 15: Sensitivity to the shrinkage threshold c in the constraint $\alpha < c$ (Avg Acc %, mean±std over 3 seeds). $c{=}1/2$ is the default used throughout the paper.

	Enzyme Reaction			EC Number		
Encoder	$c{=}1/2$	$c{=}1/3$	$c{=}1/4$	$c{=}1/2$	$c{=}1/3$	$c{=}1/4$
ESM2 650M	81.6±0.7	81.0±0.8	81.6±0.8	96.4±0.6	96.2±0.4	96.1±0.2
ESM2 150M	76.8±0.8	78.4±0.3	77.9±0.5	96.0±0.2	95.8±0.1	95.9±0.0
SaProt	78.4±0.2	77.9±1.0	78.5±0.1	95.1±0.3	94.9±0.1	94.9±0.1
ProtBert	71.8±1.4	71.7±0.7	72.5±2.1	88.8±2.0	88.8±2.1	88.7±1.9
GearNet	62.7±0.0	62.7±0.0	62.7±0.0	82.9±0.0	78.1±6.7	82.9±0.0
Avg	74.3	74.3	74.6	91.8	90.8	91.7

Table 16: Baselines with fixed hyperparameters (as in Table 1) and with per-encoder hyperparameters tuned on a held-out validation split (Avg Acc %, mean±std over 3 seeds). ProtoCIL uses no tuning.

		Enzyme Reaction (348 classes)						EC Number (4788 classes)					
Method	**HP**	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg	ESM2 650M	ESM2 150M	GearNet Edge	SaProt	ProtBert	Avg
RanPAC	fixed	74.3±0.0	70.5±0.0	21.8±0.0	73.5±0.0	65.8±0.0	61.2	82.2±0.0	79.9±0.0	26.1±0.0	84.7±0.0	79.0±0.0	70.4
	tuned	77.3±0.1	72.0±0.0	60.7±0.4	76.7±0.1	69.6±0.0	71.3	82.2±0.0	79.9±0.0	26.9±13.8	96.0±0.0	88.6±0.0	74.7
FeCAM (orig.)	fixed	61.9±0.0	49.6±0.0	35.5±0.0	60.1±0.0	38.4±0.0	49.1	82.4±0.0	77.9±0.0	61.0±0.0	77.9±0.0	67.1±0.0	73.3
	tuned	61.6±0.0	52.6±0.0	48.6±0.0	61.0±0.0	47.3±0.0	54.2	84.7±0.0	80.2±0.0	69.6±0.0	77.9±0.0	70.5±0.0	76.6
DER++	fixed	66.8±3.1	61.2±5.2	33.1±2.7	62.0±1.8	54.7±5.0	55.6	64.5±5.1	59.4±5.7	25.9±2.9	57.9±6.7	56.3±6.5	52.8
	tuned	68.4±3.6	61.4±3.4	40.1±3.1	63.0±4.1	56.2±5.8	57.8	73.0±4.6	68.0±6.9	45.2±5.8	66.8±7.4	68.2±7.2	64.2
ProtoCIL	none	81.6±0.7	76.8±0.8	62.7±0.0	78.4±0.2	71.8±1.4	74.3	96.4±0.6	96.0±0.2	82.9±0.0	95.1±0.3	88.8±2.0	91.8

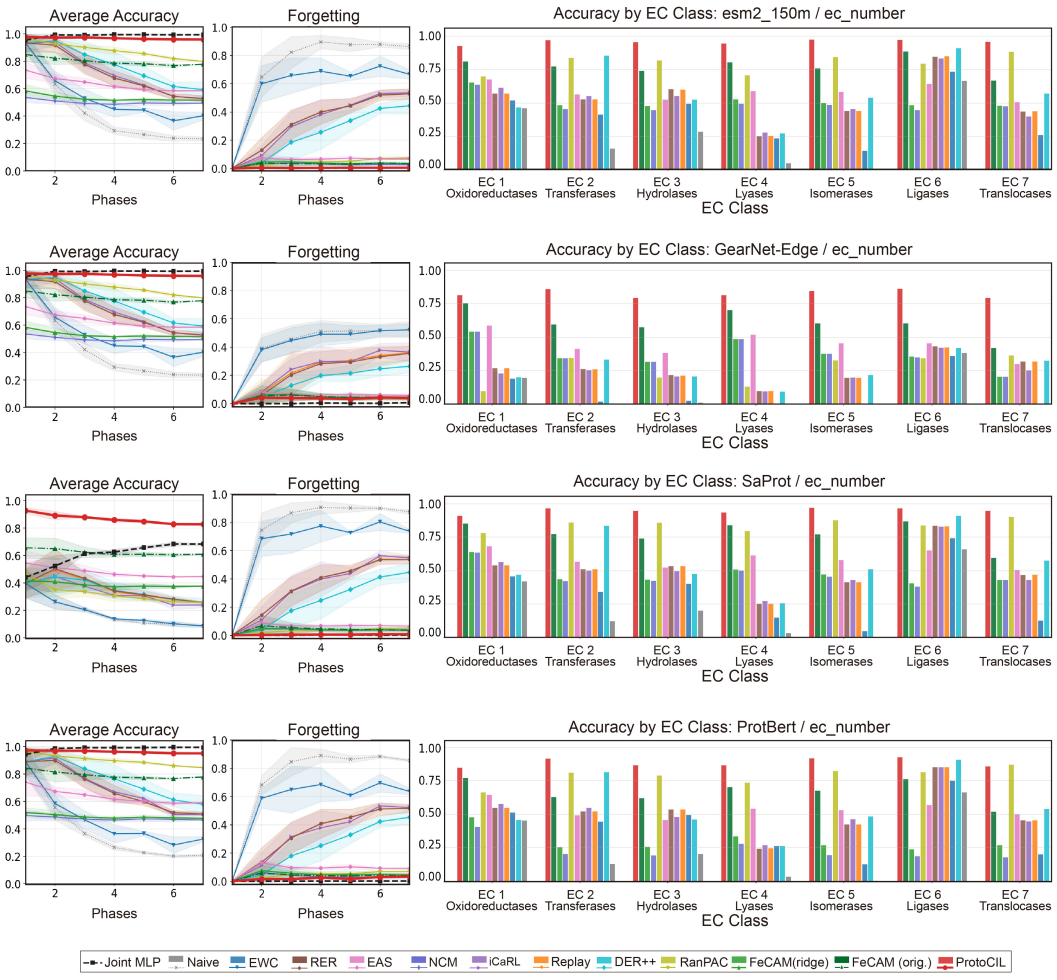


Figure 4: Per-encoder results on EC Number. **Left:** Average over phases. **Middle:** Forgetting over phases. **Right:** Final accuracy by EC class. ProtoCIL (red) maintains the highest accuracy and low forgetting across all five encoders, while parametric methods forget substantially as phases increase.

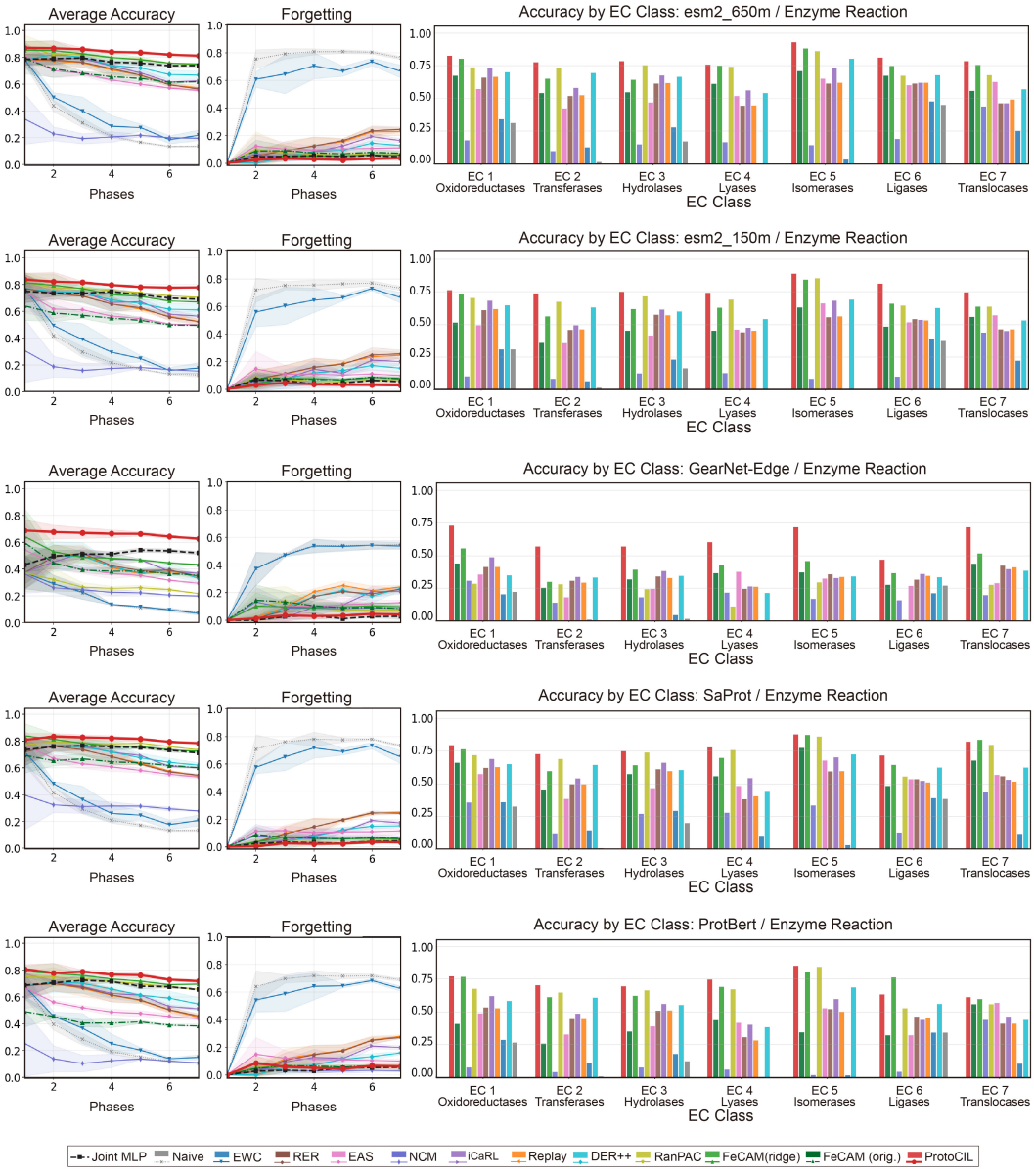


Figure 5: Per-encoder results on Enzyme Reaction. **Left:** Average over phases. **Middle:** Forgetting over phases. **Right:** Final accuracy by reaction type. ProtoCIL (red) maintains the highest accuracy and low forgetting across all five encoders, while parametric methods forget substantially as phases increase.

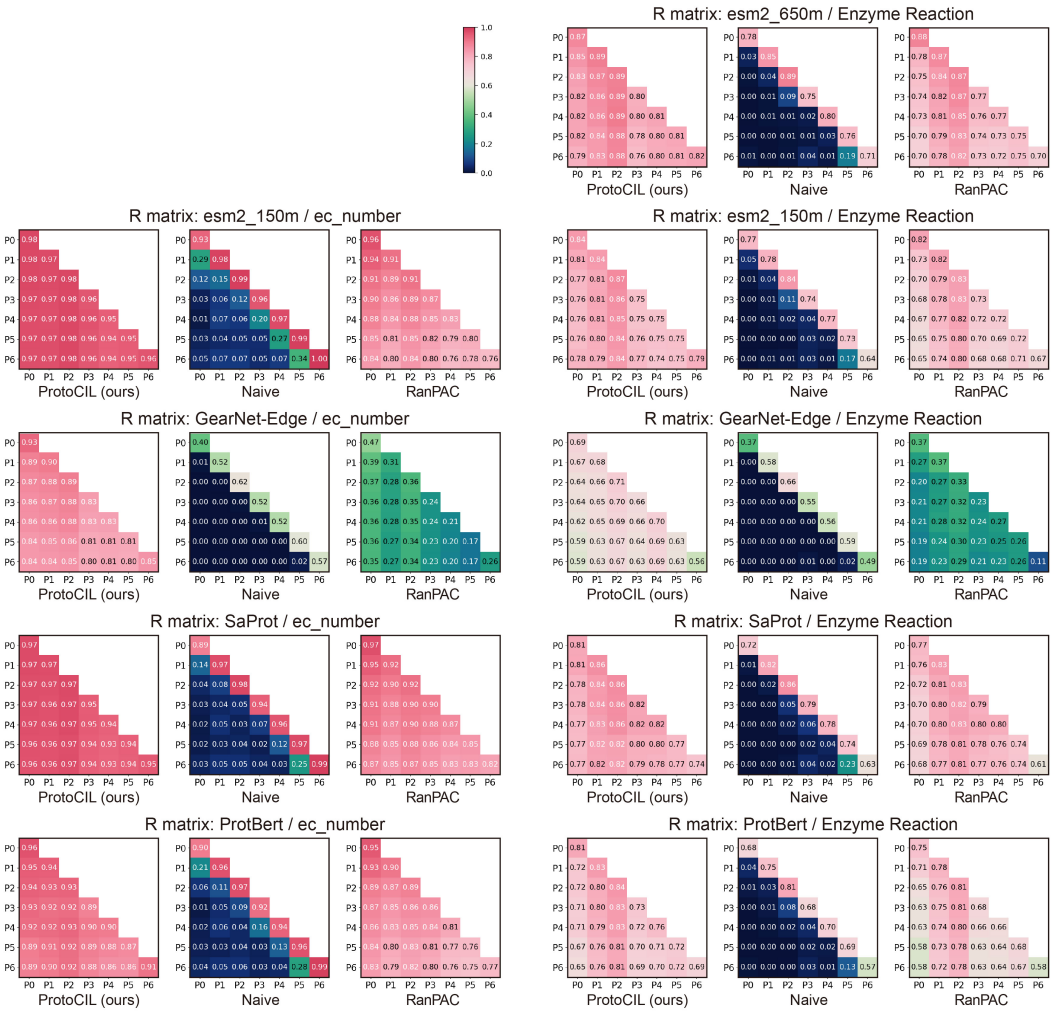


Figure 6: Phase-wise accuracy matrices for all five encoders. **Left column:** EC Number. **Right column:** Enzyme Reaction. For each encoder, ProtoCIL, Naive, and RanPAC are shown from left to right. Each entry R_{ij} is the test accuracy on phase P_i after training up to phase P_j. ProtoCIL preserves high accuracy across phases, while Naive suffers severe forgetting and RanPAC degrades gradually as classes accumulate.



Figure 7: Prototype distributions and group sizes learned by ProtoCIL on EC Number. Each panel visualizes one frozen protein encoder, where the upper subpanel shows the PC1 projection of class prototypes within each adaptive group and the lower subpanel shows the corresponding group size. The results illustrate that different encoders induce substantially different prototype geometries and group-size distributions, motivating encoder-adaptive group-wise covariance estimation rather than using a fixed grouping rule.