RELAG: A SCALABLE FRAMEWORK GENERALIZING RANDOM SPLITS TO DATA WITH LATENT RELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Random splitting can yield non-independent train–test subsets when a dataset contains related samples, as is common in certain applications such as biochemical studies. This leads to overly optimistic generalization estimates. Here, we introduce ReLaG, a modality-agnostic framework that models sample relatedness through a hierarchical latent-variable process and infers groups of related samples using proximity graphs and community detection to produce independent train–test subsets. Across molecular and protein datasets, ReLaG matches existing relation-aware methods while scaling substantially better, enabling splits at previously impractical dataset sizes. We further introduce a label-free procedure that adapts the splitting resolution to production data, aligning evaluation with the intended deployment setting. ReLaG's inferred groups provide a cheap estimate of effective dataset size, enabling diversity-aware dataset scaling.

1 INTRODUCTION

Standard machine-learning evaluation commonly relies on random train–test splits, which assume that samples are independently and identically distributed (i.i.d.) (Hastie et al., 2009). This assumption is frequently violated in real-world datasets, where samples occur in related groups, as observed in domains ranging from epidemiology and econometrics to the social sciences (O'Connell & Speiser, 2025). The issue is particularly pronounced in biochemical machine learning: evolutionary processes, combinatorial screening, and dataset bias lead to datasets that contain complex dependencies between samples (Bernett et al., 2024).

When a random split places related samples in both the training and test sets, a model can achieve high test performance by memorizing training examples rather than by learning to generalize to genuinely independent samples. This train–test leakage can therefore lead to overly optimistic generalization estimates (Patil et al., 2026). Such leakage has been documented across biochemical prediction tasks, including drug–target interaction prediction (Atas & Doğan, 2022), reaction prediction (Bradshaw et al., 2025), protein fitness prediction (Cheng et al., 2026), and binding affinity prediction (Graber et al., 2025). In some settings, a k-nearest-neighbor baseline can even rival more complex predictors (Mathai & Kirchmair, 2020). This illustrates why random splits can reward memorization of local similarity rather than generalization to independent samples.

Consequently, benchmark performance may not reflect performance on production data. This discrepancy may contribute to the generalization gaps reported in several areas of biochemical machine learning, including protein–ligand affinity modeling, TCR–peptide binding, pMHC prediction, and DNA-encoded-library screening (Brown, 2025; Grazioli et al., 2022; Marzella et al., 2024; Dolorfino et al., 2026). Relation-aware splitting methods address this issue by keeping related samples within the same split (train or test), thereby preventing leakage. However, their use is currently limited by modality-specific assumptions, by a computational cost that prevents application to moderate to large datasets, and, for many, by a similarity threshold defining dependence for which no selection method exists, leaving users to fall back on conventional values that may not suit the task at hand (Teufel et al., 2023; Steshin, 2023).

In this work, we introduce ReLaG (**Re**lation-based **La**tent **G**rouping), a generic and modality-agnostic framework for relation-aware dataset splitting. We formalize a two-level latent-variable model in which observed samples are conditioned on group-specific latent variables, such that samples sharing the same latent variable form a group of related samples. From this model, we derive a

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.

practical algorithm that infers these groups from observed data. Given a modality-specific distance function, ReLaG constructs an approximate proximity graph, identifies communities as estimates of these groups, and treats each community as an indivisible unit during random partitioning. By assigning entire communities to a single subset, ReLaG prevents related samples from occurring on both sides of a subset boundary, making both subsets independent under the latent-variable model.

Using an approximate proximity graph, ReLaG runs the most expensive step of the algorithm in $O(n \log n)$ time rather than the $O(n^2)$ cost of previous approaches that rely on pairwise comparisons (Joeres et al., 2025; Fernández-Díaz et al., 2024a). Across molecular and protein datasets, it yields minimal leakage comparable to existing methods (Joeres et al., 2025; Fernández-Díaz et al., 2024a) while scaling more favorably for larger datasets, enabling relation-aware splits at scale.

The proximity graph depends on a distance threshold setting the resolution at which samples are considered related. We propose a method to select the most promising threshold from unlabeled production data as the resolution at which production and training data are most similar at the community level. Without relational structure, ReLaG reduces to random splitting, thereby generalizing it. Conversely, when no threshold aligns them, the production data are detected as out of distribution.

The community structure additionally provides an approximate effective dataset size, N_{eff}, defined by the number of communities. Across our experiments, N_{eff} is associated with generalization performance than the raw number of samples. This suggests that N_{eff} could support more efficient dataset scaling by prioritizing experiments expected to yield new communities.

Our contributions are as follows:

- We formalize a modality-independent generative process for datasets containing related samples, providing a common foundation for relation-aware splitting across domains.
- We derive ReLaG, an approach for partitioning any datasets generated under this process at scale by avoiding the quadratic cost of pairwise comparisons.
- We introduce a label-free threshold-selection procedure that adapts the splitting resolution to the intended production data.
- We show that ReLaG's community structure provides a useful estimate of effective dataset diversity with potential to guide more efficient dataset scaling.

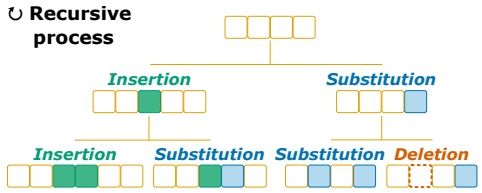
Although we focus on biochemical datasets, where sample relatedness is particularly salient and measurable, the underlying formalism applies to any domain with related samples that can be described by the proposed generative process, such as social networks or recommendation systems.

2 Related Work

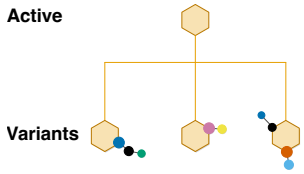
Relation-aware splitting methods can be broadly grouped into optimization-based, clustering-based, and graph-based approaches. Optimization-based methods define an explicit criterion for a desirable split. For molecular data, SIMPD uses a multi-objective genetic algorithm to construct splits that mimic temporal drift in medicinal chemistry (Landrum et al., 2023), whereas MOOD selects, among existing splitting protocols, the one whose train–test distance distributions best match a target deployment setting (Tossou et al., 2024). DataSAIL formulates the problem more generally as a combinatorial optimization problem that minimizes cross-subset similarity (Joeres et al., 2025).

Clustering-based approaches first cluster similar samples in the Euclidean space and assign entire clusters to subsets. For molecules, hierarchical clustering splits apply single-linkage or HDBSCAN clustering to ECFP4 fingerprints (Bai et al., 2021); alternative work instead proposes UMAP-based clustering (Guo et al., 2024). For biological sequences, SpanSeq combines alignment-free k-mer distances with DBSCAN clustering and constrained cluster assignment (Ferrer Florensa et al., 2024), while Pfam-based splits assign protein families to folds (Zhu et al., 2022).

Graph-based methods partition an explicit similarity graph. Lo-Hi formulates molecular dataset split as a balanced vertex k-cut problem (Steshin, 2023). For biological sequences, dataset are partitioned along connected components of a thresholded similarity graph in binding affinity prediction (Kanakala et al., 2023) and by AutoPeptideML (Fernández-Díaz et al., 2024b). QMAP further subdivides large connected components through Leiden community detection (Lavertu et al., 2026).



(a) **Evolution:** sequences diverge from common ancestors via insertions, substitutions and deletions.



(b) **Fragment based design:** drug hits are refined into close variants sharing a core scaffold.

Figure 1: Illustration of two generative processes that produce datasets containing related samples.

GraphPart constructs a weighted sequence-similarity graph and iteratively reassigns or removes samples until no cross-subset edge exceeds a specified threshold (Teufel et al., 2023). Hestia generalizes the GraphPart framework beyond sequences to multiple data modalities and out-of-distribution prediction settings (Fernández-Díaz et al., 2024a). Structural-interface similarity graphs have also been used to reveal leakage in protein–protein interaction benchmarks (Bushuiev et al., 2024). Graph-based methods rely on a similarity threshold defining which samples are connected, which is typically set by convention.

Among these, DataSAIL and Hestia are the only generic methods designed to operate across modalities, but they pursue different objectives. DataSAIL minimizes similarity between subsets (Joeres et al., 2025), whereas Hestia retrains a model across a threshold sweep to estimate performance under increasing train–test dissimilarity (Fernández-Díaz et al., 2024a), an informative but costly aggregate compared to a single split. ReLaG instead starts from a modality-independent generative model of relatedness and produces, at scale, subsets that are independent under it, at a resolution inferred from unlabeled production data to align evaluation with the deployment setting.

3 Problem Formulation

Generative process. Many datasets arise from generative processes in which common underlying sources produce multiple related observations. Biochemical data provide clear examples of this (Fig. 1). Biological sequences, such as proteins, peptides, or genes, descend from ancestral sequences through evolution (Nei & Kumar, 2000), such that sequences sharing a recent common ancestor tend to be close in edit distance (Fig. 1a) (Bouchard-Côté & Jordan, 2012). Similarly, in bioactive molecule discovery, chemists synthesize local variants around active hits or combinatorial scaffolds, producing families of structurally related molecules (Fig. 1b) (Vamathevan et al., 2019; Peterson & Liu, 2023). Although evolution and molecular screening operate on different objects, they induce the same dataset-level structure: samples are organized into groups of related observations rather than being independently and identically distributed.

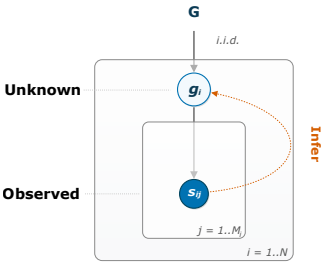


Figure 2: **Two-level latent-variable model.** Observed samples s_{ij} are generated conditionally on group-specific latent variables g_i, which are drawn i.i.d. from a fixed population-level distribution G. The objective is to infer this group structure from the observed samples.

A latent variable model of relatedness. We formalize the shared structure induced by these generative processes using a two-level latent-variable model (Fig. 2). Let g_i denote an unobserved group-specific latent variable, drawn independently from a population-level distribution G. The jth sample from group i is then drawn from a distribution conditioned on g_i:

$$g_i \overset{\text{i.i.d.}}{\sim} G, \qquad s_{ij} \mid g_i \sim p(s \mid g_i).$$

Samples sharing the same g_i are dependent when marginalizing over g_i, while samples from different groups are independent, so the i.i.d. assumption only holds at the group level. For instance, a group may correspond to a combinatorial molecular library, variants of a drug hit, a family or superfamily of proteins (Sonnhammer et al., 1997). The appropriate grouping thus depends on the resolution at which the problem is posed, which we make explicit through a resolution parameter τ and write $g_i \overset{\text{i.i.d.}}{\sim} G(\tau)$.

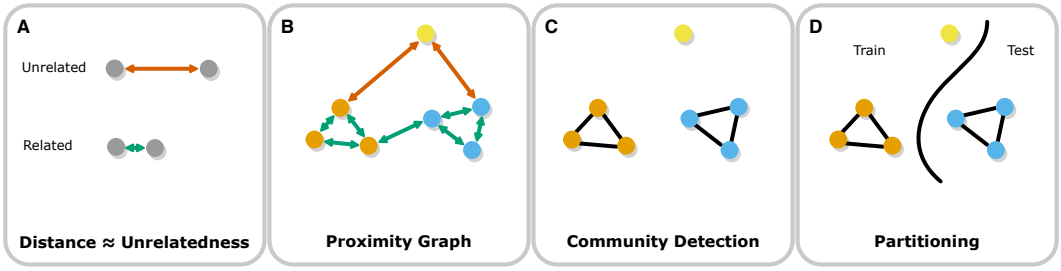


Figure 3: **ReLaG algorithm overview.** **(A)** The algorithm requires a distance function that correlates with the probability that two samples are unrelated. **(B)** A proximity graph is built using this distance function. **(C)** Communities are identified, then graph edges that are unlikely to represent relationships are removed. Each community is an estimate of a group, i.e., a set of samples conditioned on the same latent variable g_i. **(D)** The dataset is partitioned along the community boundaries.

4 Proposed Methods

4.1 The ReLaG Algorithm

The objective of ReLaG is to infer, from the observed data, groups of related samples that are conditioned on the same latent variable g_i (dotted orange arrows in Fig. 2). We refer to these inferred groups as *communities*. Since groups are mutually independent under the model, each community is treated as an atomic unit: all of its samples must be assigned to the same split (e.g., training, validation, or test). A standard random split can then be applied over communities rather than over individual samples, preventing relations from crossing subset boundaries. It is achieved in four steps: defining a distance function, constructing an approximate proximity graph, identifying communities through community detection, and finally, partitioning (Fig. 3).

Distance function. The method is modality-agnostic in that it requires only a distance function appropriate for the data. Consider two samples s_{ij} and s_{kl}, belonging to groups i and k, respectively. We require a distance d for which proximity is informative of shared group membership:

$$P\big(i \neq k \mid s_{ij}, s_{kl}\big) = f(d(s_{ij}, s_{kl}))\,,$$

where f is non-decreasing. Thus, larger distances correspond to a higher probability that two samples originate from different groups, and the smaller the distance, the more probable they belong in the same group (Fig. 3A).

Approximate proximity graph. Given a distance threshold τ, we define a proximity graph whose nodes are samples and whose edges connect pairs satisfying $d(s_{ij}, s_{kl}) < \tau$ (Fig. 3B). Constructing this graph exactly requires $n(n-1)/2$ distance evaluations and therefore an $O(n^2)$ time complexity. This can be prohibitive for large datasets or expensive distance functions. We instead construct an approximate proximity graph using the Hierarchical Navigable Small World (HNSW) data structure (Malkov & Yashunin, 2020). HNSW organizes samples into exponentially sparser graph layers, allowing for approximate nearest-neighbor insertion and retrieval in $O(\log n)$ time. Standard HNSW is optimized for nearest-neighbor retrieval speed and therefore stores only a sparse subset of the pairwise relationships encountered during graph construction. However, many additional sample pairs are compared while searching for neighbors, and subsequently discarded from the HNSW structure. For proximity-graph construction, these evaluations remain informative: any evaluated pair satisfying $d(s_{ij}, s_{kl}) < \tau$ provides evidence of a candidate relation. We therefore augment the standard HNSW construction by retaining all evaluated pairs whose distance falls below τ as proximity-graph edges, rather than only those selected as HNSW edges. This yields a denser approximation of the proximity graph without requiring additional distance evaluations (Fig. 3B).

Community detection. The distance function provides only a noisy proxy for group membership: samples from different groups can nevertheless fall within the distance threshold by chance, thereby inducing spurious proximity-graph edges. Under a null model in which proximity arises by chance between unrelated samples, an edge occurs with probability $p_0 = P(d(s_{ij}, s_{kl}) < \tau \mid i \neq k)$, independently of group membership. Consequently, such edges are uniformly dispersed across groups. In contrast, samples from the same group are more likely to lie within the threshold and therefore

induce an excess of within-group edges. Thus, groups manifest as regions of the proximity graph whose internal connectivity exceeds that expected under the null model.

We estimate p_0 using a modality- and problem-specific reference procedure that removes genuine relatedness while preserving the marginal data distribution, for example, shuffling residues within sequences, sampling random valid molecules, or pairing images from distinct classes. This procedure yields the distance distribution of unrelated pairs from which p_0 can be estimated. We then apply the Leiden community-detection algorithm (Traag et al., 2019) with the Constant Potts Model (CPM) objective (Traag et al., 2011): $\sum_c \left[e_c - \gamma\binom{n_c}{2}\right]$, where e_c and n_c are the number of internal edges and nodes in community c, respectively, and $\gamma = p_0$ is the null edge probability. Maximizing this objective identifies communities whose internal edge density exceeds that expected under the null model (Fig. 3C). When a modality-specific null model is unavailable, communities can be obtained using a traditional Modularity objective (Traag et al., 2019).

Community-level partitioning. Finally, we treat each community as an atomic unit and apply a random train–test split over communities rather than individual samples. Consequently, all samples within a community are assigned to the same subset, resulting in independent subsets under the latent-variable model (Fig. 3D).

4.2 Selecting the Distance Threshold

The chosen distance threshold τ has a large impact on the obtained communities. For some data modalities, such as biological sequences, the latent-variable model is resolution-dependent: the same dataset can admit valid groupings at multiple levels of relatedness. In ReLaG, this resolution is controlled by the distance threshold τ used to construct the proximity graph. Larger values of τ connect more samples and yield coarser groups, whereas smaller values produce finer groups. Since the true resolution is unobserved, finding the right threshold for the task at hand is challenging.

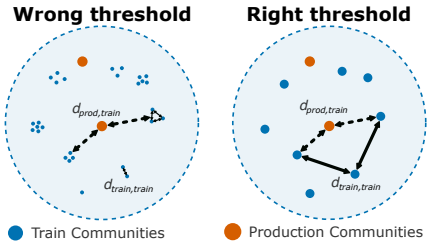


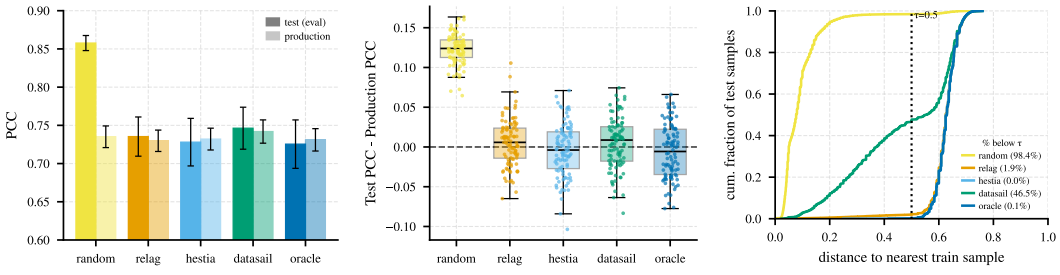
Figure 4: **Threshold selection intuition.** A wrong threshold makes train communities closer or further away from each other than they are from production communities. The right threshold makes production communities integrate nicely in the community distribution.

In the following, we propose an automated scheme to select the most promising threshold $\tau^\star$ using unlabeled production data, i.e., samples on which the trained model is expected to be deployed. Such samples are often available before labels are collected. For instance, in hit discovery, candidate molecules can be sampled from the screening library before any assay is run. The hypothesis is that the production samples are drawn from the same distribution as the training data. Here, training data refer to the full dataset to be split prior to partitioning. Under this hypothesis, the appropriate resolution is the one at which production and training samples are indistinguishable at the group level, as illustrated in Figure 4. Intuitively, $\tau^\star$ should make the distribution of distances among training communities match the distribution between production and training communities.

We search for $\tau^\star$ by sweeping over candidate thresholds. To keep the sweep tractable, we build a single HNSW graph over the union of training and production samples and use its sparse base layer as a candidate graph. For each candidate threshold τ, we retain edges shorter than τ, estimate the null model, and apply Leiden community detection with the CPM objective. This yields three sets of communities: $\mathcal{C}^\tau_{\text{train}}$ denotes the set of *pure-training* communities (containing only training samples); $\mathcal{C}^\tau_{\text{prod}}$ denotes *pure-production* communities (only production samples); and $\mathcal{C}^\tau_{\text{hybrid}}$ denotes *hybrid* communities (samples from both sets). For each threshold, we compare how far pure-training and pure-production communities lie from the training data. We measure the distance between two communities A and B by their single-linkage distance, i.e., the smallest distance between any sample of one and any sample of the other, $d_{\text{SL}}(A,B) = \min_{a\in A,\, b\in B} d(a,b)$. We then consider the empirical cumulative distribution functions of single-linkage distances, from every pure-training and from every pure-production community to every other training-containing community:

$$F^\tau_{\text{train}\to\text{train}}(x) = \tfrac{1}{n_1}\left|\left\{d_{\text{SL}}(A,B) \le x \;:\; A \in \mathcal{C}^\tau_{\text{train}},\; B \in \mathcal{C}^\tau_{\text{train}} \cup \mathcal{C}^\tau_{\text{hybrid}}, A \neq B\right\}\right|, \quad (1)$$

$$F^\tau_{\text{prod}\to\text{train}}(x) = \tfrac{1}{n_2}\left|\left\{d_{\text{SL}}(A,B) \le x \;:\; A \in \mathcal{C}^\tau_{\text{prod}},\; B \in \mathcal{C}^\tau_{\text{train}} \cup \mathcal{C}^\tau_{\text{hybrid}}\right\}\right|, \quad (2)$$



(a) Mean Pearson correlation (PCC) and std between $\hat{y}$ and y on the test split and on production data.

(b) Per-repeat difference between test and production PCC. 0 indicates a faithful evaluation.

(c) ECDFs of the distance ($1-$ global identity) from each test sample to its nearest training sample.

Figure 5: **Relation-aware splits yield evaluations that match production performance.** In each of 100 repeats, we sample independently two synthetic peptide datasets (15,000 sequences, 1,000 families each) under the latent-variable model: an annotated dataset to split and a production dataset.

with n_1 and n_2 chosen such that $F^{\tau}_{\text{train}\to\text{train}}(\infty)=F^{\tau}_{\text{prod}\to\text{train}}(\infty)=1$. We compare these distributions using the Kolmogorov-Smirnov statistic (Massey, 1951) $\text{KS}(F_1, F_2) = \max_x |F_1(x)-F_2(x)|$ and select the τ that minimizes it: $\tau^\star = \arg\min_\tau \text{KS}\left(F^{\tau}_{\text{train}\to\text{train}}, F^{\tau}_{\text{prod}\to\text{train}}\right)$.

The procedure returns $\tau^\star$ together with the minimal KS statistic, which measures how closely production and training data can be aligned. A value near 0 indicates that the hypothesis holds at resolution $\tau^\star$ while a value near 1 indicates that no threshold can align the two distributions, signaling that the production data are out of distribution. The full procedure is given in Algorithm 1.

5 Evaluation

5.1 Synthetic validation

We compare ReLaG with DataSAIL (Joeres et al., 2025) and Hestia (Fernández-Díaz et al., 2024a), the two existing relation-aware splitting methods designed to operate across data modalities, and with a random-splitting baseline. Assessing whether a split yields a faithful estimate of generalization requires a labeled production dataset, which benchmark datasets lack. We therefore first evaluate on synthetic data. As reference point, we include an oracle that splits along true groups.

Setup. We instantiate the latent-variable model with a peptide generative process in which each family descends from an independently sampled ancestral sequence through recursive mutations (Appendix). Labels are given by a deterministic function of the sequence corrupted by Gaussian noise. For each of 100 repeats, we draw two independent datasets from this process: an annotated dataset to be partitioned and a production dataset. The threshold selection procedure yields $\tau^\star = 0.5$ (Figure 11b), which we use for all threshold-based methods and at which ReLaG communities recover the ground-truth families almost exactly (the average Adjusted Rand Index is 0.991 ± 0.006).

Test versus production. For each split, we encode peptides using ESM Cambrian (Candido et al., 2026), train a two-layer MLP prediction head on the training subset, and measure its Pearson correlation (PCC) between predictions $\hat{y}$ and targets y on both the test subset and the production dataset. Random splitting overestimates production performance substantially with a test PCC of 0.86 compared to 0.74 in production (Figure 5a). In contrast, the test performance of every relation-aware method closely matches its production performance. Per repeat, the difference between test and production PCC is centered on zero for ReLaG ($+0.005$), Hestia (-0.004), DataSAIL ($+0.004$), and the oracle (-0.006) (Figure 5b).

Train–test distance. The source of this overestimation is visible in the distance from each test sample to its nearest training sample (Figure 5c). Under random splitting, 98.4% of test samples lie within $\tau^\star$ of a training sample, i.e., nearly every test sample has a relative in the training set. A model can thus achieve high test performance by exploiting similarity to training samples rather than learning the underlying sequence–property relationship, an advantage that vanishes on unseen families (Mathai & Kirchmair, 2020). ReLaG (1.9%) and Hestia (0.0%) nearly match the oracle

(0.1%), whereas DataSAIL leaves 46.5% of test samples within τ of the training set. Because DataSAIL minimizes overall train–test similarity rather than enforcing a specific threshold, it does not guarantee that individual test samples lack close relatives in training. Together, these results show that random splitting overestimates generalization when samples are related, whereas relation-aware splits yield test performance that reflects production performance. ReLaG performs on par with existing relation-aware methods.

5.2 Real Data

We next evaluate the same methods on real data, comprising six molecular and two protein datasets. Although these datasets lack production data, we examine whether they exhibit the same train–test distance and test performance patterns as the synthetic data.

Train–test distance. For a controlled comparison on real data, we use a common distance threshold for all datasets within each modality: 0.5 for proteins and 0.6 for molecules. As on synthetic data, random splitting leaves most test samples within the threshold of a training sample (79–82%), whereas ReLaG yields residual similarity comparable to Hestia and lower than DataSAIL across both modalities (Figure 14). For proteins, ReLaG places only 1% of test samples below the threshold, compared to 10% for Hestia (Figure 14b). For molecules, Hestia places no test samples below the threshold, compared to 23% for ReLaG (Figure 14a). This difference reflects distinct objectives. Hestia removes all train–test similarity below a chosen threshold, explicitly constructing an out-of-distribution test set. ReLaG instead targets statistical independence under the latent-variable model, allowing residual similarity between samples of different groups that fall within the threshold by chance.

Test performance. We follow the same protocol as for synthetic data, encoding proteins with ESM Cambrian (Candido et al., 2026) and molecules with ChemBERTa (Chithrananda et al., 2020), and repeat each experiment over 10 split seeds. Random splitting yields higher test performance than every relation-aware method, consistent with the overestimation observed on synthetic data, whereas ReLaG, Hestia, and DataSAIL yield comparable performance (Figure 6). Although production performance cannot be measured on these datasets, relation-aware methods agree as they do on synthetic data, where their test estimates matched production.

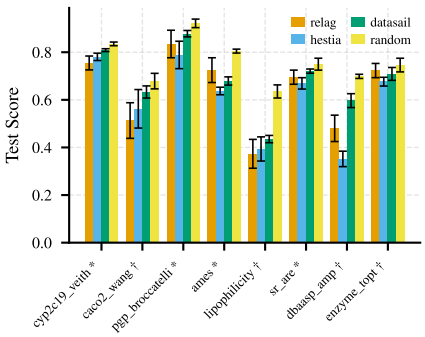


Figure 6: **Performances on real data.** Downstream MLP test performance of dataset-splitting strategies across eight datasets (six molecular and two protein datasets). Results are averaged over 10 split seeds, with error bars denoting standard deviations. We report AUROC for classification tasks and Pearson correlation coefficient for regression tasks.

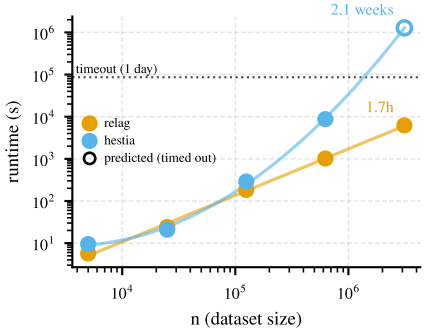
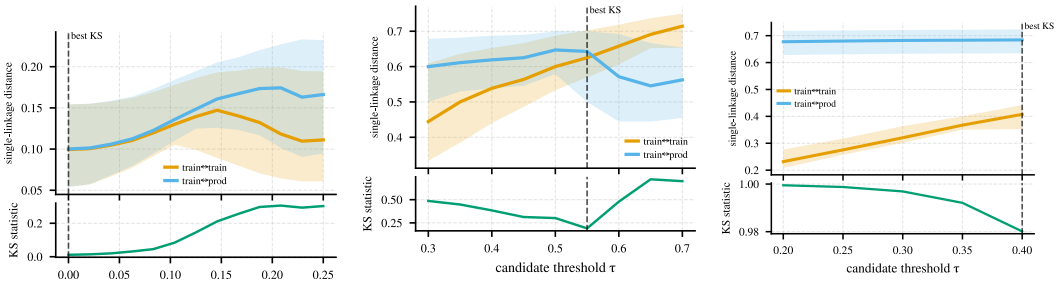


Figure 7: **Scaling on PeptideAtlas subsets.** Runtime as a function of dataset size for ReLaG and Hestia, the two fastest compared methods. Runs were limited to one day; values above the dashed line are extrapolated. Log–log fits indicate approximately linear scaling for ReLaG and quadratic scaling for Hestia ($R^2 = 0.998$ for both).

Scalability. Beyond evaluation quality, the computational cost of existing relation-aware methods limits their application to large datasets. We therefore compare the runtime of ReLaG and Hestia, the two fastest of the studied methods, on increasingly large subsets of PeptideAtlas (Desiere et al., 2006). Runs exceeding one day are terminated, and their runtimes are extrapolated from fitted scaling curves. As shown in Figure 7, the methods have comparable runtime for datasets containing tens of thousands of samples, but ReLaG scales more favorably as dataset size increases. At $n = 3.125$M, Hestia exceeds the one-day time budget, with an extrapolated runtime of 2.1 weeks, whereas ReLaG completes in 1.7 hours, an estimated 207× speedup that enables relation-aware splitting at



(a) i.i.d.: MNIST.

(b) Related: DBAASP (training) vs. PeptideAtlas (Production).

(c) OOD: BELKA training set vs. non-triazine test subset.

Figure 8: **Threshold selection characterizes the relatedness regime.** Top: single-linkage distances from pure-training and pure-production communities to training communities (15th–85th percentile shaded). Bottom: KS statistic.

previously impractical dataset scales. We observe a similar trend on the molecule modality (Appendix Figure 10a).

5.3 Threshold

We next evaluate the threshold-selection procedure across three relatedness regimes. For data without detectable relational structure, we expect the selected threshold to be close to zero since all groups are singletons. For relational data, the procedure should select a non-zero threshold corresponding to a meaningful grouping resolution. Finally, when production data are out-of-distribution (OOD), no threshold should align the train–train and production–train distance distributions.

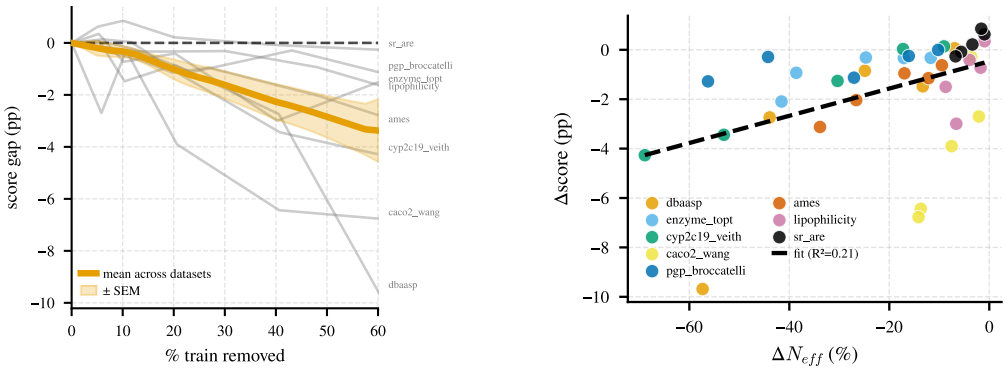
We consider three datasets illustrating these regimes (Fig. 8). MNIST (LeCun et al., 1998) provides an approximately i.i.d. setting. By treating the test set as production data, the KS statistic is minimized at $\tau = 0$ (Fig. 8a), consistent with the absence of relational structure. DBAASP (Pirtskhalava et al., 2021), a dataset of antimicrobial peptides with substantial sequence relatedness, represents the relational regime. Using the PeptideAtlas database (Desiere et al., 2006) as production data, the threshold converges at $\tau = 0.55$ (Fig. 8b), indicating a recoverable relational grouping. Finally, BELKA (Quigley et al., 2024), a DNA-encoded-library screening benchmark released through a NeurIPS 2024 Kaggle competition, tests the OOD regime: a post-competition analysis found that none of the ∼2,000 participating teams outperformed random predictions on the designated OOD test subset (Dolorfino et al., 2026). Its training molecules all contain a triazine core absent from all molecules of that test subset, and although training samples become densely connected by $\tau = 0.4$, no threshold aligns the two distance distributions (minimum KS statistic of 0.98; Fig. 8c), consistent with the production samples being OOD relative to training.

These regimes illustrate that ReLaG can be seen as generalizing random splitting along the relatedness axis. When no relational structure is detected, each sample forms its own group, and group-level random splitting recovers conventional random splitting. When groups are present, ReLaG partitions along their inferred boundaries. At the opposite extreme, when all samples belong to a single group, no non-leaking split is possible, and ReLaG instead signals that the training and production datasets are mutually out-of-distribution.

5.4 Free diversity estimate

Our formalism and algorithm provide a natural estimate of the effective dataset size: N_{eff}, corresponding to the number of communities. We evaluate whether N_{eff} is more informative of downstream generalization than the raw dataset size N across our benchmark datasets.

Since most analyzed benchmarks have no production data, we choose for each dataset a threshold at which community sizes follow a realistic heavy-tailed distribution as too small thresholds fragment the data into mostly small communities. We then remove either complete communities or an equal number of randomly selected samples from the training set. Removing entire communities produces a larger performance decrease than removing the same number of samples at random (Fig. 9a),



(a) **Removing random samples vs communities** The test-score difference between removing samples uniformly at random and complete communities is shown as a function of the fraction of training data removed. Thin lines denote individual datasets; the thick line and shaded region denote the mean and SEM, respectively. Negative values indicate that removing entire communities produces a larger performance decrease.

(b) **Effective dataset size.** Each point shows the difference in test performance when dropping entire train communities compared to dropping train samples at random, ensuring the training set sizes are equal across both conditions. The x-axis shows the resulting difference in N_{eff}, the number of train communities, and the y-axis the difference in test score (community - random). Larger losses in N_{eff} are associated with larger performance drops (r = 0.46).

Figure 9: Community structure captures an aspect of training-set diversity that is not reflected by the raw sample count and is associated with generalization performance.

indicating that the loss of entire communities is more harmful to downstream performance than a comparable reduction in raw sample count. Across datasets, the performance gap is correlated with reductions in N_{eff} (Fig. 9b). This pattern is consistent with within-group redundancy limiting the contribution of related samples to generalization (Bayley & Hammersley, 1946).

The BELKA dataset provides an illustrative limiting case. As discussed previously, because the production set is OOD, the training set's proximity graph collapses to a complete graph (i.e., every sample is connected to every other sample) before the two distance distributions can align. This yields $N_{\text{eff}} = 1$ and indicates that the training data alone do not provide independent groups from which to generalize to other groups such as the test one. Doing so would require increasing the number of groups, thus diversity.

More broadly, these results suggest a strategy for more efficient dataset scaling: prioritize the acquisition and annotation of samples that form new communities, rather than samples that primarily increase the size of existing ones.

6 Conclusion

We introduced a relation-aware splitting framework derived from a latent-variable model of sample relatedness. By inferring groups through an approximate proximity graph and community detection, ReLaG constructs train–test splits that preserve inferred group boundaries. Across molecular and protein datasets, it provides leakage-aware evaluations comparable to existing generic methods while enabling substantially larger-scale splits. A label-free threshold-selection procedure further adapts the splitting resolution to production data, recovering random splitting when no relational structure exists and flagging out-of-distribution production data. Its communities additionally provide an effective dataset size, N_{eff}, which may support diversity-aware dataset scaling.

Although ReLaG is fast relative to other relation-aware methods, it remains slower than random splitting and is therefore unnecessary when samples are known to be independent. The method also requires a meaningful distance function, which may be non-trivial to define for some modalities. Threshold selection further requires unlabeled production data, which are not always available. Finally, without a modality-specific null model, ReLaG falls back to the less interpretable modularity objective. Future work will further explore the generality of ReLaG by applying its framework beyond biochemical data.

7 AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical equations, design and provide feedback on research experiments. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets and assist in the writing of proofs, are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, create or edit software code, draft parts of a research paper, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature and propose a title and keywords for a research paper. We have reviewed all AI-assisted work. For each paper identified, we verified that it supported the cited claims and conducted traditional literature searches to minimize missed publications. For software package development, we reviewed and tested all AI-generated code to ensure it functioned as expected and met our quality standards. For research experiments, we verified that the code accurately implemented the methodology and critically assessed the robustness and scientific validity of the experimental design. We critically evaluated any scientific feedback provided by generative AI models rather than accepting it without verification. Mathematical equations were independently verified by at least two co-authors to ensure their mathematical soundness. For figure generation, AI was used to assist in writing the code, while we independently verified that the figures accurately represented the underlying data. Finally, any text reformulated by AI was reviewed by the authors for accuracy and fidelity to the original content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

8 REPRODUCIBILITY STATEMENT

We provide the complete implementation of ReLaG as an open-source Rust library with Python bindings, together with all code needed to reproduce our results, at `https://anonymous.4open.science/r/relag-D24C`. The code automatically downloads and preprocesses the datasets, runs all experiments, and generates every figure in the paper. The repository README lists the scripts and commands for reproducing each experiment and figure.

REFERENCES

Heval Atas and Tunca Doğan. How to Approach Machine Learning-based Prediction of Drug/Compound-Target Interactions. bioRxiv, 2022. Pages: 2022.05.01.490207 Section: New Results.

Peizhen Bai, Filip Miljković, Yan Ge, Nigel Greene, Bino John, and Haiping Lu. Hierarchical Clustering Split for Low-Bias Evaluation of Drug-Target Interaction Prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 641–644, 2021. doi: 10.1109/BIBM52615.2021.9669515.

G. V. Bayley and J. M. Hammersley. The "Effective" Number of Independent Observations in an Autocorrelated Time Series. *Supplement to the Journal of the Royal Statistical Society*, 8(2): 184–197, 1946. doi: 10.2307/2983560.

Judith Bernett, David B. Blumenthal, Dominik G. Grimm, Florian Haselbeck, Roman Joeres, Olga V. Kalinina, and Markus List. Guiding questions to avoid data leakage in biological machine learning applications. *Nature Methods*, 21(8):1444–1453, 2024. doi: 10.1038/s41592-024-02362-y.

H. G. Boman, D. Wade, I.A. Boman, B. Wåhlin, and R.B. Merrifield. Antibacterial and antimalarial properties of peptides that are cecropin-melittin hybrids. *FEBS Letters*, 259(1):103–106, 1989. doi: 10.1016/0014-5793(89)81505-4.

A. Bouchard-Côté and Michael I. Jordan. Evolutionary inference via the Poisson Indel Process. *Proceedings of the National Academy of Sciences*, 110:1160–1166, 2012. doi: 10.1073/pnas.1220450110.

John Bradshaw, Anji Zhang, Babak Mahjour, David E. Graff, Marwin H. S. Segler, and Connor W. Coley. Challenging Reaction Prediction Models to Generalize to Novel Chemistry. *ACS Central Science*, 11(4):539–549, 2025. doi: 10.1021/acscentsci.5c00055.

Fabio Broccatelli, Emanuele Carosati, Annalisa Neri, Maria Frosini, Laura Goracci, Tudor I. Oprea, and Gabriele Cruciani. A Novel Approach for Predicting P-Glycoprotein (ABCB1) Inhibition Using Molecular Interaction Fields. *Journal of Medicinal Chemistry*, 54(6):1740–1751, 2011. doi: 10.1021/jm101421d.

Benjamin P. Brown. A generalizable deep learning framework for structure-based protein–ligand affinity ranking. *Proceedings of the National Academy of Sciences*, 122(42):e2508998122, 2025. doi: 10.1073/pnas.2508998122.

Anton Bushuiev, Roman Bushuiev, Jiri Sedlar, Tomas Pluskal, Jiri Damborsky, Stanislav Mazurenko, and Josef Sivic. Revealing data leakage in protein interaction benchmarks. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language Modeling Materializes a World Model of Protein Biology. bioRxiv, 2026. ISSN: 2692-8205 Pages: 2026.06.03.729735 Section: New Results.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Le Song. xTrimoPGLM: unified 100-billion-parameter pretrained transformer for deciphering the language of proteins. *Nature Methods*, 22(5):1028–1039, 2025. doi: 10.1038/s41592-025-02636-z.

Zongrui Cheng, Haoxin Wu, and Dengming Ming. Beyond Random Splits: Assessing the Generalization of Graph and Vector Models for WT-Structure-Only Drug Resistance Prediction under Protein-Disjoint Evaluation. *Computational and Structural Biotechnology Journal*, 35(1):0144, 2026. doi: 10.34133/csbj.0144.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. arXiv, 2020. arXiv:2010.09885 [cs.LG].

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Frank Desiere, Eric W. Deutsch, Nichole L. King, Alexey I. Nesvizhskii, Parag Mallick, Jimmy Eng, Sharon Chen, James Eddes, Sandra N. Loevenich, and Ruedi Aebersold. The PeptideAtlas project. *Nucleic Acids Research*, 34(suppl_1):D655–D658, 2006. doi: 10.1093/nar/gkj040.

Marissa Dolorfino, Daniel Santos Perez, Yao Fu, Shu-Hang Lin, Sean McCarty, Matthew J. O'Meara, and Terra Sztain. Assessing the Generalizability of Machine Learning and Physics-Based Methods with DNA-Encoded Libraries. bioRxiv, 2026. ISSN: 2692-8205 Pages: 2026.04.18.719394 Section: New Results.

David Eisenberg, Robert M. Weiss, Thomas C. Terwilliger, and William Wilcox. Hydrophobic moments and protein structure. *Faraday Symposia of the Chemical Society*, 17:109–120, 1982. doi: 10.1039/FS9821700109.

Raul Fernández-Díaz, Hoang Thanh Lam, Vanessa López, and Denis C. Shields. A new framework for evaluating model out-of-distribution generalisation for the biochemical domain. In *The Thirteenth International Conference on Learning Representations*, 2024a.

Raúl Fernández-Díaz, Rodrigo Cossio-Pérez, Clement Agoni, Hoang Thanh Lam, Vanessa Lopez, and Denis C Shields. AutoPeptideML: a study on how to build more trustworthy peptide bioactivity predictors. *Bioinformatics*, 40(9):btae555, 2024b. doi: 10.1093/bioinformatics/btae555.

Alfred Ferrer Florensa, Jose Juan Almagro Armenteros, Henrik Nielsen, Frank Møller Aarestrup, and Philip Thomas Lanken Conradsen Clausen. SpanSeq: similarity-based sequence data splitting method for improved development and assessment of deep learning projects. *NAR Genomics and Bioinformatics*, 6(3):lqae106, 2024. doi: 10.1093/nargab/lqae106.

David Graber, Peter Stockinger, Fabian Meyer, Siddhartha Mishra, Claus Horn, and Rebecca Buller. Resolving data bias improves generalization in binding affinity prediction. *Nature Machine Intelligence*, 7(10):1713–1725, 2025. doi: 10.1038/s42256-025-01124-5.

Filippo Grazioli, Anja Mösch, Pierre Machart, Kai Li, Israa Alqassem, Timothy J. O'Donnell, and Martin Renqiang Min. On TCR binding predictors failing to generalize to unseen peptides. *Frontiers in Immunology*, 13, 2022. doi: 10.3389/fimmu.2022.1014256.

Qianrong Guo, Saiveth Hernandez-Hernandez, and Pedro J. Ballester. Scaffold Splits Overestimate Virtual Screening Performance. In *Artificial Neural Networks and Machine Learning – ICANN 2024*, pp. 58–72. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72359-9_5.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. doi: 10.1073/pnas.89.22.10915.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf H. Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Ruili Huang, Menghang Xia, Dac-Trung Nguyen, Tongan Zhao, Srilatha Sakamuru, Jinghua Zhao, Sampada A. Shahane, Anna Rossoshek, and Anton Simeonov. Tox21Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways as Mediated by Exposure to Environmental Chemicals and Drugs. *Frontiers in Environmental Science*, 3, 2016. doi: 10.3389/fenvs.2015.00085.

Roman Joeres, David B. Blumenthal, and Olga V. Kalinina. Data splitting to avoid information leakage with DataSAIL. *Nature Communications*, 16(1):3337, 2025. doi: 10.1038/s41467-025-58606-8.

Ganesh Chandan Kanakala, Rishal Aggarwal, Divya Nayar, and U. Deva Priyakumar. Latent Biases in Machine Learning Models for Predicting Binding Affinities Using Popular Data Sets. *ACS Omega*, 8(2):2389–2397, 2023. doi: 10.1021/acsomega.2c06781.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (CASP)—Round XV. *Proteins: Structure, Function, and Bioinformatics*, 91(12):1539–1549, 2023. doi: 10.1002/prot.26617.

Gregory A. Landrum, Maximilian Beckers, Jessica Lanini, Nadine Schneider, Nikolaus Stiefl, and Sereina Riniker. SIMPD: an algorithm for generating simulated time splits for validating machine learning approaches. *Journal of Cheminformatics*, 15(1):119, 2023. doi: 10.1186/s13321-023-00787-9.

Anthony Lavertu, Jacques Corbeil, and Pascal Germain. QMAP: a benchmark for standardized evaluation of antimicrobial peptide MIC and hemolytic activity regression. *Scientific Reports*, 16 (1):25387, 2026. doi: 10.1038/s41598-026-56004-8.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Chris-Andre Leimeister, Jendrik Schellhorn, Svenja Dörrer, Michael Gerth, Christoph Bleidorn, and Burkhard Morgenstern. Prot-SpaM: fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 8(3):giy148, 2019. doi: 10.1093/gigascience/giy148.

Yu A. Malkov and D. A. Yashunin. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

Dario F. Marzella, Giulia Crocioni, Tadija Radusinović, Daniil Lepikhov, Heleen Severin, Dani L. Bodor, Daniel T. Rademaker, ChiaYu Lin, Sonja Georgievska, Nicolas Renaud, Amy L. Kessler, Pablo Lopez-Tarifa, Sonja I. Buschow, Erik Bekkers, and Li C. Xue. Geometric deep learning improves generalizability of MHC-bound peptide predictions. *Communications Biology*, 7(1): 1661, 2024. doi: 10.1038/s42003-024-07292-1.

Frank J. Massey. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. doi: 10.2307/2280095.

Neann Mathai and Johannes Kirchmair. Similarity-Based Methods and Machine Learning Approaches for Target Prediction in Early Drug Discovery: Performance and Scope. *International Journal of Molecular Sciences*, 21(10):3585, 2020. doi: 10.3390/ijms21103585.

Alex T Müller, Gisela Gabernet, Jan A Hiss, and Gisbert Schneider. modlAMP: Python for antimicrobial peptides. *Bioinformatics*, 33(17):2753–2755, 2017. doi: 10.1093/bioinformatics/btx285.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. doi: 10.1016/0022-2836(70)90057-4.

Masatoshi Nei and Sudhir Kumar. *Molecular Evolution and Phylogenetics*. Oxford University Press, 2000. doi: 10.1093/oso/9780195135848.001.0001.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In *Advances in Neural Information Processing Systems*, pp. 64331–64379, 2023. doi: 10.52202/075280-2810.

Nathaniel Sean O'Connell and Jaime Lynn Speiser. OpenClustered: an R package with a benchmark suite of clustered datasets for methodological evaluation and comparison. *BMC Medical Research Methodology*, 25(1):92, 2025. doi: 10.1186/s12874-025-02548-8.

Prathamesh Patil, Arpit Jain, and Aswanth Krishnan. Beyond the Performance Illusion: Structure-Aware Stratified Partitioning and Curriculum Distributionally Robust Optimization for Spatially Correlated Domains. arXiv, 2026. arXiv:2607.02055.

Alexander A. Peterson and David R. Liu. Small-molecule discovery through DNA-encoded libraries. *Nature Reviews Drug Discovery*, 22(9):699–722, 2023. doi: 10.1038/s41573-023-00713-6.

Malak Pirtskhalava, Anthony A Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, and Michael Tartakovsky. DBAASP v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids Research*, 49(D1):D288–D297, 2021. doi: 10.1093/nar/gkaa991.

Ian K. Quigley, Andrew Blevins, Brayden J. Halverson, and Nate Wilkinson. BELKA: The Big Encoded Library for Chemical Assessment. In *NeurIPS 2024 Competition Track*, 2024.

David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Erik L.L. Sonnhammer, Sean R. Eddy, and Richard Durbin. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function, and Bioinformatics*, 28(3):405–420, 1997. doi: 10.1002/(SICI)1097-0134(199707)28:3⟨405::AID-PROT10⟩3.0.CO;2-L.

Simon Steshin. Lo-Hi: Practical ML Drug Discovery Benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Felix Teufel, Magnús Halldór Gíslason, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. GraphPart: homology partitioning for biological sequence analysis. *NAR Genomics and Bioinformatics*, 5(4):lqad088, 2023. doi: 10.1093/nargab/lqad088.

Prudencio Tossou, Cas Wognum, Michael Craig, Hadrien Mary, and Emmanuel Noutahi. Real-World Molecular Out-Of-Distribution: Specification and Investigation. *Journal of Chemical Information and Modeling*, 64(3):697–711, 2024. doi: 10.1021/acs.jcim.3c01774.

V. A. Traag, P. Van Dooren, and Y. Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1):016114, 2011. doi: 10.1103/PhysRevE.84.016114.

V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019. doi: 10.1038/s41598-019-41695-z.

Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019. doi: 10.1038/s41573-019-0024-5.

Henrike Veith, Noel Southall, Ruili Huang, Tim James, Darren Fayne, Natalia Artemenko, Min Shen, James Inglese, Christopher P. Austin, David G. Lloyd, and Douglas S. Auld. Comprehensive Characterization of Cytochrome P450 Isozyme Selectivity across Chemical Libraries. *Nature biotechnology*, 27(11):1050–1055, 2009. doi: 10.1038/nbt.1581.

Ning-Ning Wang, Jie Dong, Yin-Hua Deng, Min-Feng Zhu, Ming Wen, Zhi-Jiang Yao, Ai-Ping Lu, Jian-Bing Wang, and Dong-Sheng Cao. ADME Properties Evaluation in Drug Discovery: Prediction of Caco-2 Cell Permeability Using a Combination of NSGA-II and Boosting. *Journal of Chemical Information and Modeling*, 56(4):763–773, 2016. doi: 10.1021/acs.jcim.5b00642.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/c7sc02664a.

Congying Xu, Feixiong Cheng, Lei Chen, Zheng Du, Weihua Li, Guixia Liu, Philip W. Lee, and Yun Tang. In silico Prediction of Chemical Ames Mutagenicity. *Journal of Chemical Information and Modeling*, 52(11):2840–2847, 2012. doi: 10.1021/ci300400a.

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Hui Zhu, Jincai Yang, and Niu Huang. Assessment of the Generalization Abilities of Machine-Learning Scoring Functions for Structure-Based Virtual Screening. *Journal of Chemical Information and Modeling*, 62(22):5485–5502, 2022. doi: 10.1021/acs.jcim.2c01149.

A Appendix

A.1 Additional Methods

A.1.1 Null Model

Estimating the null edge probability,

$$p_0(\tau) = P(d(s_{ij}, s_{kl}) < \tau \mid i \neq k)\,,$$

corresponding to the probability that two unrelated samples fall within distance τ, is modality- and problem-specific. We therefore estimate the null distance distribution using Monte Carlo procedures designed to remove relational structure while preserving the marginal characteristics of the data.

For protein sequences, we sample 10 million random pairs from PeptideAtlas (Desiere et al., 2006). For each sequence, we independently shuffle its amino acids before computing pairwise distances. This procedure destroys sequence homology while preserving sequence length and amino-acid composition, thereby retaining the marginal sequence distribution while removing relational structures. For molecules, directly shuffling molecular fingerprints would destroy chemical structure and produce distances substantially larger than those observed between real molecules, yielding an unrealistically small estimate of p_0. Instead, we generate $100,000$ chemically valid molecules using randomly generated SELFIES (Krenn et al., 2020), with molecular sizes sampled from $\mathcal{N}(50, 5)$, and compute distances for 10 million random pairs. For MNIST, we construct the null distribution from a subset of ImageNet (Deng et al., 2009) by downscaling images to the MNIST resolution and converting them to grayscale. 10 million random pairs are sampled from distinct ImageNet classes.

The probability $p_0(\tau)$ may correspond to a rare event, for which direct Monte Carlo estimation can be unreliable when few or no sample pairs satisfy $d < \tau$. We therefore extrapolate the lower 1% tail by fitting a Generalized Pareto Distribution (GPD) to the smallest 1% of sampled distances and estimate $p_0(\tau)$ from the fitted distribution. The resulting estimate is used as the CPM resolution parameter, i.e.,

$$\gamma = p_0(\tau).$$

A.1.2 Data Preparation

Datasets. We evaluate ReLaG on eight real datasets, including six molecular datasets and two sequence-based datasets (Table 1). For all datasets, we discard the predefined train–test splits and re-split the pooled data from scratch.

The molecular datasets include five single-task datasets from the Therapeutics Data Commons (TDC) (Huang et al., 2021): `cyp2c19_veith`, which measures inhibition of the CYP2C19 enzyme (Veith et al., 2009), `caco2_wang`, which measures Caco-2 permeability (Wang et al., 2016), `pgp_broccatelli`, which measures P-glycoprotein inhibition (Broccatelli et al., 2011), `ames`, which measures Ames mutagenicity (Xu et al., 2012), and `lipophilicity`, which measures AstraZeneca logD (Wu et al., 2018). We additionally use the SR-ARE task from the multi-task Tox21 dataset (Huang et al., 2016). For sequence-based datasets, we use DBAASP antimicrobial peptides (Pirtskhalava et al., 2021), retaining only sequences composed of canonical L-amino acids with an *E. coli* MIC label, with labels expressed as $\log_{10}$ MIC obtained from the QMAP benchmark package (Lavertu et al., 2026). We also use the enzyme optimal-temperature regression dataset (Chen et al., 2025).

Synthetic dataset. Additionally, we generate synthetic peptide datasets that follow the two-level latent-variable model, where each group is a family of sequences descending from a common ancestor. Each dataset contains $15{,}000$ sequences from $1{,}000$ families. For each family, we generate a random ancestral sequence: its length is drawn from the empirical length distribution of PeptideAtlas sequences of 20 to 50 residues (Desiere et al., 2006), and each residue is drawn independently from the amino-acid frequencies of PeptideAtlas. Families are then grown by a branching process: starting from the ancestor, each sequence produces a Poisson-distributed number of descendants (mean 1.6), each obtained by applying $1 + \mathrm{Poisson}(2)$ random edits to its parent. An edit is a substitution (85%), an insertion (7.5%), or a deletion (7.5%) at a uniformly sampled position, with substitutions drawn from the BLOSUM62 conditional distribution (Henikoff & Henikoff, 1992) and insertions from the background frequencies. The accumulated number of edits from the ancestor is capped at 10, and growth proceeds breadth-first until the target dataset size is reached. The annotated and production datasets are generated with different random seeds and therefore share no family.

The target is a non-linear function of three physicochemical descriptors computed with modlAMP (Müller et al., 2017): the net charge at pH 7, the maximal Eisenberg hydrophobic moment over 11-residue windows (Eisenberg et al., 1982), and the Boman index (Boman et al., 1989). Let z_c, z_m and z_b denote the standardized charge, log hydrophobic moment and Boman index, and

$s(x) = \tanh(0.7x)/0.7$ a soft saturation. The target is

$$y = f(x) + \varepsilon,$$

where

$$f(x) = s(z_m) + 0.8\, s(z_c) + s(z_m)\, s(z_c) - s(z_m)\, s(z_b) + 0.8 \sin(1.5\, z_b).$$

f is standardized to unit variance and $\varepsilon \sim \mathcal{N}(0, 0.05^2)$. The standardization constants are fixed once, so annotated and production datasets share the same function. Since y depends only on the sequence, any systematic difference between test and production performance is attributable to the split.

Filtering. Filtering is performed once before splitting so that all methods and downstream models use identical sample indices. Molecular samples with missing SMILES or labels, or with unparsable SMILES, are removed. Protein samples with missing or non-finite labels are removed. We retain sequences composed of the 20 standard amino acids, along with `X` in `enzyme_topt`, which is supported by ESM Cambrian. For PeptideAtlas (Desiere et al., 2006), which is used only for null-model estimation and scalability experiments, we merge all build FASTA files, deduplicate sequences, and retain sequences of length at most 100. For BELKA, we remove duplicates and extract the non-triazine subset of the test set as the OOD dataset.

Representations. For ReLaG, molecules are represented using 2048-bit Morgan fingerprints with radius 2 (Rogers & Hahn, 2010), computed with RDKit. Protein sequences are represented directly as amino-acid strings. For downstream prediction, molecules are encoded using ChemBERTa (`seyonec/ChemBERTa-zinc-base-v1`) (Chithrananda et al., 2020), using the `[CLS]` representation from the final hidden layer. Proteins are encoded using ESM Cambrian 300M (Candido et al., 2026), using mean-pooled final-layer residue embeddings. The encoders are frozen, and only the prediction head is trained. Distances used are $1-$ Tanimoto similarity on Morgan fingerprints for molecules, $1-$ global alignment identity (Needleman & Wunsch, 1970) for protein and peptide sequences, ProtSpaM for UniRef50 (A.3.2) (Leimeister et al., 2019), and $1-$ cosine similarity for MNIST.

Table 1: Datasets used in the experiments. N is the number of samples after filtering.

Dataset	Modality	N	Task	Metric	Encoder
`cyp2c19_veith`	molecule	12,665	classification	AUROC	ChemBERTa
`ames`	molecule	7,278	classification	AUROC	ChemBERTa
`sr_are`	molecule	5,832	classification	AUROC	ChemBERTa
`lipophilicity`	molecule	4,200	regression	PCC	ChemBERTa
`pgp_broccatelli`	molecule	1,218	classification	AUROC	ChemBERTa
`caco2_wang`	molecule	910	regression	PCC	ChemBERTa
`dbaasp`	peptide	10,664	regression	PCC	ESM-C 300M
`enzyme_topt`	protein	1,886	regression	PCC	ESM-C 300M

A.1.3 Split Comparison

Protocol. All splitting methods operate on the same data, representations, and use the same MLP architecture. The only difference between methods is how the train–test split is constructed. We allocate 80% of the samples to the training and validation set and 20% to the test set. Of the 80%, 10% is randomly held out as a validation set for early stopping. We evaluate regression tasks using the Pearson correlation coefficient between $\hat{y}$ and y (PCC) and classification tasks using AUROC. Each experiment is repeated over 10 random seeds (1–10) for each method and dataset.

Threshold. We use fixed thresholds within each modality to ensure a controlled comparison across splitting methods. Because the benchmark datasets contain no explicit production samples, the threshold-selection procedure cannot be applied. We therefore use a fixed threshold per modality: $\tau = 0.60$ for molecular datasets and $\tau = 0.50$ for protein datasets with ReLaG. Equivalently, the corresponding thresholds for Hestia are 0.40 Tanimoto similarity for molecules and 50% sequence identity for proteins. These thresholds are kept fixed across datasets within each modality to ensure a fair comparison.

Prediction head. For downstream evaluation, we use a two-layer MLP on top of the frozen representations. Given an encoder output of dimension d, the architecture is

$$\texttt{Linear}(d, 2d) \to \texttt{ReLU} \to \texttt{Linear}(2d, 1).$$

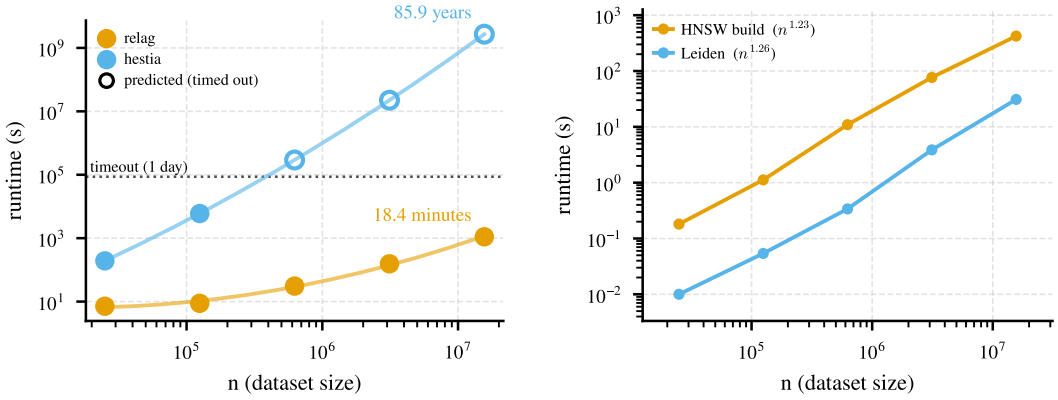
The model is optimized with Adam using a learning rate of 10^{-3} and a batch size of 64 for up to 300 epochs. Early stopping is performed using the validation loss with a patience of 15 epochs, and the checkpoint with the best validation loss is restored before evaluation. We use mean squared error for regression and binary cross-entropy with logits for classification.

Train–test distance. To quantify residual relatedness between the train and test sets, we compute the distance from each test sample to its nearest training sample using an exact brute-force search over the full training set. We use the same distance function as the corresponding splitting method: Tanimoto distance for molecular fingerprints and $1-$ global sequence alignment identity (Needleman & Wunsch, 1970) for protein sequences. We report the empirical cumulative distribution function (ECDF) of these nearest-neighbor distances.

A.2 Scaling

We extend our scaling investigation to the molecule modality (Fig. 10a). Although ReLaG scales substantially more favorably than Hestia, its empirical scaling appears slightly super-linear, above the expected $O(n \log n)$.

To investigate this discrepancy, we decompose the runtime of ReLaG by algorithmic step and measure the contribution of each component as the dataset size increases (Fig. 10b). This figure shows that the Leiden community detection seems to increase faster than the build process. This is expected because its computational complexity scales with the number of edges in the graph, which scales faster than $O(n \log n)$, suggesting that at very large scale, the runtime will be dominated by the community detection step.



(a) Runtime against dataset size on BELKA subsets, for ReLaG and Hestia. Runs are capped at one day, and values above the dashed line are extrapolated from the fitted curve.

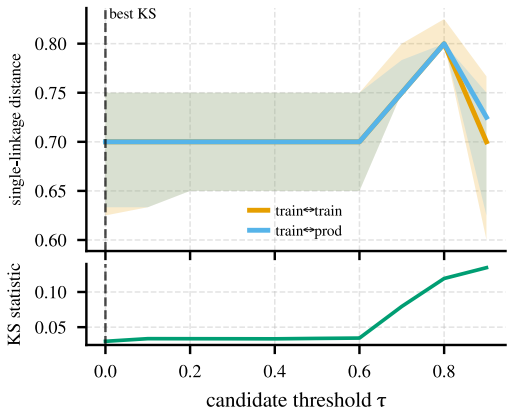
(b) Per-stage breakdown of the ReLaG pipeline on BELKA subsets. Building the HNSW dominates the total runtime at every size, but community detection grows faster ($n^{1.26}$ against $n^{1.23}$).

Figure 10: Scaling of ReLaG on molecular modality. Both axes are logarithmic. Exponents in the legend are fitted on the log–log space.

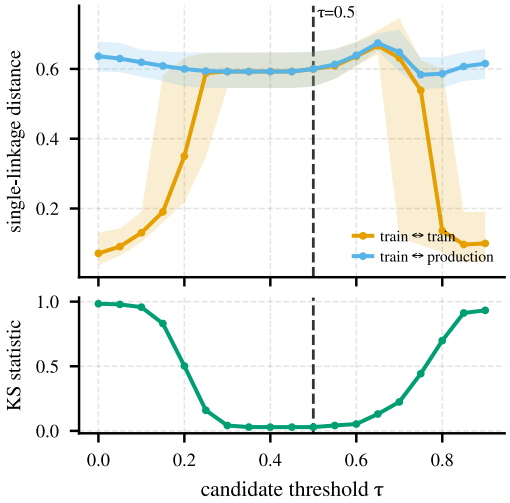
A.3 Threshold Selection

A.3.1 Algorithm

Algorithm 1 details the threshold-selection procedure. Without prior knowledge of a suitable range, we sweep τ upward from 0, where every sample is a singleton. The sweep stops when every candidate edge has been retained, as larger thresholds then leave the graph unchanged, or when no



(a) UniRef50, with ProteinGym, PEER and CASP15 as production data. The KS statistic is minimized at $\tau^\star = 0$ and remains nearly flat for $0 < \tau \lesssim 0.6$.



(b) Synthetic peptides, with an independently generated production dataset. The KS statistic is minimized over a plateau, and the largest threshold on it, $\tau^\star = 0.5$, is selected.

Figure 11: **Threshold selection on large-scale and synthetic data.** Upper panels: single-linkage distances from pure-training communities (train↔train) and from pure-production communities (train↔production) to training-containing communities, with shading denoting the 15th–85th percentile range. Lower panels: the corresponding Kolmogorov–Smirnov statistic. Dashed lines mark the selected threshold $\tau^\star$.

pure-training or pure-production communities remain, as the distance distributions are then undefined. Among the evaluated thresholds, we select the one minimizing the KS statistic, taking the largest such threshold in case of ties.

A.3.2 UNIREF EXPERIMENT

We next apply the threshold-selection procedure at a scale representative of the training data used by state-of-the-art protein language models. We use the complete UniRef50 dataset, comprising approximately 38.7 million protein sequences. The production dataset is constructed as the union of three widely used protein benchmarks for evaluating protein language models: ProteinGym (Notin et al., 2023), PEER (Xu et al., 2022), and CASP15 (Kryshtafovych et al., 2023).

At this scale, we replace global sequence alignment with the ProtSpaM kernel (Leimeister et al., 2019) for distance computation. Global alignment has $O(L^2)$ complexity with respect to sequence length and is already the dominant computational cost on our smaller protein datasets, making its application to UniRef50 intractable. ProtSpaM instead estimates sequence distance from spaced-word matches, with approximately linear complexity in sequence length, making the threshold search feasible at this scale. This substitution does not substantially alter the interpretation of the distance: ProtSpaM and global alignment distances agree closely on the same sequence pairs (Fig. 12), allowing thresholds expressed in ProtSpaM distance to be interpreted in terms of sequence identity.

The KS statistic reaches its minimum at $\tau = 0$ and remains essentially flat after, for $\tau \in (0, 0.6]$. It then increases for larger thresholds, reaching 0.080 at $\tau = 0.7$ and 0.135 at $\tau = 0.9$. The procedure therefore selects $\tau = 0$.

This result is expected for two reasons. First, UniRef50 is clustered at 50% sequence identity, equivalent to 0.6 in ProtSpaM distance (See Fig 12), so relationships up to $\tau = 0.6$ have already been removed by construction. Second, the production benchmarks have a similar distance to the training set as the training set is to itself (Table 2). The production-to-training distribution is closer at the lower tail, with a first percentile of 0.00 compared with 0.13 for train-to-training, and remains

Algorithm 1 Threshold selection

Require: Dataset to split $\mathcal{S}$, unlabeled production data $\mathcal{P}$, distance function d, increasing candidate thresholds $0 = \tau_1 < \tau_2 < \cdots < \tau_K$
Ensure: Selected threshold $\tau^\star$ and minimal KS statistic $\mathrm{KS}^\star$

$E \leftarrow$ base-layer edges of an HNSW graph built over $\mathcal{S} \cup \mathcal{P}$ with distance d
for $\tau \in (\tau_1, \ldots, \tau_K)$ **do**
  $E_\tau \leftarrow \{(u, v) \in E : d(u, v) < \tau\}$ ▷ Proximity graph at resolution τ
  if $E_\tau = E$ **then**
    break ▷ Complete graph
  end if
  $\gamma \leftarrow p_0(\tau)$ ▷ Null edge probability
  $\mathcal{C}^\tau \leftarrow \textsc{Leiden-CPM}(\mathcal{S} \cup \mathcal{P}, E_\tau, \gamma)$
  $\mathcal{C}^\tau_{\mathrm{train}} \leftarrow \{A \in \mathcal{C}^\tau : A \subseteq \mathcal{S}\}$ ▷ Pure-training
  $\mathcal{C}^\tau_{\mathrm{prod}} \leftarrow \{A \in \mathcal{C}^\tau : A \subseteq \mathcal{P}\}$ ▷ Pure-production
  $\mathcal{C}^\tau_{\mathrm{hybrid}} \leftarrow \mathcal{C}^\tau \setminus (\mathcal{C}^\tau_{\mathrm{train}} \cup \mathcal{C}^\tau_{\mathrm{prod}})$ ▷ Hybrid
  if $\mathcal{C}^\tau_{\mathrm{train}} = \emptyset$ **or** $\mathcal{C}^\tau_{\mathrm{prod}} = \emptyset$ **then**
    break ▷ Distributions undefined
  end if
  Compute $F^{(\tau)}_{\mathrm{train}\to\mathrm{train}}$ and $F^{(\tau)}_{\mathrm{prod}\to\mathrm{train}}$
  $k_\tau \leftarrow \mathrm{KS}\big(F^{(\tau)}_{\mathrm{train}\to\mathrm{train}}, F^{(\tau)}_{\mathrm{prod}\to\mathrm{train}}\big)$
end for
$\mathrm{KS}^\star \leftarrow \min_\tau k_\tau$
$\tau^\star \leftarrow \max\{\tau : k_\tau = \mathrm{KS}^\star\}$ ▷ Largest threshold on ties
return $\tau^\star, \mathrm{KS}^\star$

closer in the upper tail at the 90th and 99th percentiles. Across the majority of the distribution, the two are very similar.

Thus, at small resolutions, the training and production samples are consistent with the same relatedness distribution, leading ReLaG to correctly select $\tau = 0$ and fall back to random splitting. This is consistent with the intended behavior of ReLaG, as UniRef50 is pre-clustered to remove sequence redundancy, mitigating related-sequence leakage under random splitting. By contrast, in the BELKA challenge, the OOD production samples were too distant from the training distribution for any threshold to align the two distributions. These results suggest that PLM pretraining operates in a high-effective-sample-size regime: in our experiment, $N_{\mathrm{eff}} \approx N$ relative to widely used protein benchmarks, providing a potential explanation for why they generalize to multiple target tasks. BELKA represents the opposite regime, where aligning with the OOD production data requires a resolution at which N_{eff} collapses, providing a potential explanation for the poor generalization observed across models (Dolorfino et al., 2026).

Table 2: Nearest-train-neighbor distance on UniRef50: percentiles of the distance from a training sequence to its closest other training sequence (train↔train), and from a production benchmark sequence to its closest training sequence (train↔prod).

Percentile	train↔train	train↔prod	prod − train
1	**0.127**	**0.000**	$\mathbf{-0.127}$
5	0.500	0.516	$+0.016$
10	0.600	0.600	0.000
25	0.650	0.650	0.000
50	0.700	0.700	0.000
75	0.750	0.750	0.000
90	**0.775**	**0.750**	$\mathbf{-0.025}$
95	0.800	0.800	0.000
99	**0.825**	**0.817**	$\mathbf{-0.008}$

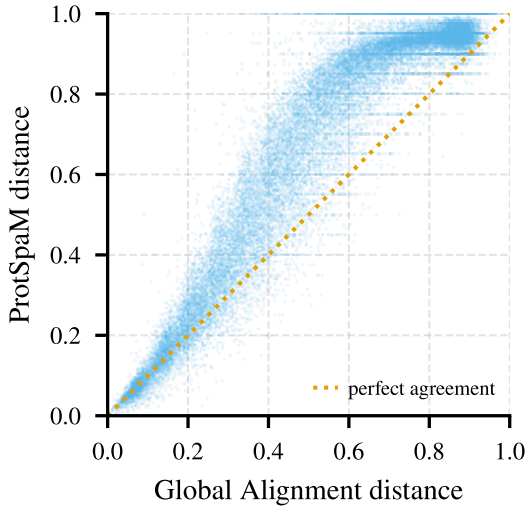


Figure 12: ProtSpaM distance against global alignment distance, on 39,380 pairs built from real UniRef50 seeds mutated at a sweep of rates (BLOSUM62 substitutions and indels) so the alignment axis is covered end to end. The two are monotonically related (Pearson $r = 0.954$, Spearman $r = 0.911$).

A.4 Singletons: Noise or Signal?

We investigate whether singleton samples contribute useful information or instead behave as outliers that reduce predictive performance. For each dataset, we train one MLP using only singleton training samples and a second MLP using an equally sized random sample of non-singleton training samples. Both models use the same embeddings, architecture, and the same fixed validation set obtained by a ReLaG split, so the only difference is the subset of training samples used. We repeat the comparison over five splits with six repetitions per split and assess significance using a two-sided t-test with $p < 0.05$.

To identify meaningful communities, thresholds are selected separately for each dataset using the heavy-tailed community-size heuristic of Section 5.4. We use the modularity objective for all datasets, avoiding the need to recompute the CPM resolution parameter for each threshold and task.

The results reveal three regimes (Table 3). First, models trained on singletons outperform community-matched ones in `caco2_wang`, `sr_are`, and `lipophilicity`. This is consistent with singletons being genuine groups of size one: each would then carry independent information, whereas samples within a community share information with their relatives. Second, singletons are less informative in `ames`, `dbaasp`, and `cyp2c19_veith` since models trained on singletons perform worse. This suggests that these singleton samples behave as outliers with a mapping that is less aligned with the prediction task. Finally, `pgp_broccatelli` and `enzyme_topt` show no statistically significant differences between singleton and community-matched samples, consistent with singletons representing a mixture of informative and uninformative samples.

These results indicate that whether singletons provide signal or noise is a property of the dataset rather than of the splitting method. We therefore exclude singletons from N_{eff} except when the statistical test indicates that they provide significantly greater per-sample signal than a community-matched sample, suggesting that they represent genuine groups rather than noise. Under this criterion, singletons contribute to N_{eff} for `caco2_wang`, `sr_are`, and `lipophilicity`, while they are excluded for the remaining five datasets (Fig. 9b).

Table 3: Singleton information content. Comparison of models trained on singleton samples and equally sized samples from non-singleton communities, evaluated on the same validation set. Δ is the performance difference (singleton $-$ community), with $\Delta > 0$ indicating greater information content in singletons. Mean over 5 splits $\times$ 6 repeats. p-values are from two-sided t-tests. Rows are grouped by higher, lower, or indistinguishable singleton performance.

Dataset	Metric	τ	Singleton	Community	Δ	p
caco2_wang	PCC	0.6	0.473 ± 0.122	0.417 ± 0.103	$+0.056$	1.1×10^{-2}
lipophilicity	PCC	0.4	0.540 ± 0.037	0.500 ± 0.054	$+0.040$	2.6×10^{-5}
sr_are	AUROC	0.4	0.710 ± 0.030	0.682 ± 0.030	$+0.029$	1.5×10^{-6}
cyp2c19_veith	AUROC	0.5	0.779 ± 0.045	0.796 ± 0.020	-0.017	3.8×10^{-2}
dbaasp	PCC	0.3	0.483 ± 0.061	0.520 ± 0.041	-0.037	1.3×10^{-5}
ames	AUROC	0.7	0.530 ± 0.075	0.640 ± 0.064	-0.109	8.8×10^{-5}
pgp_broccatelli	AUROC	0.6	0.817 ± 0.095	0.830 ± 0.055	-0.013	0.382
enzyme_topt	PCC	0.8	0.573 ± 0.103	0.602 ± 0.108	-0.029	0.175

A.5 Additional figures

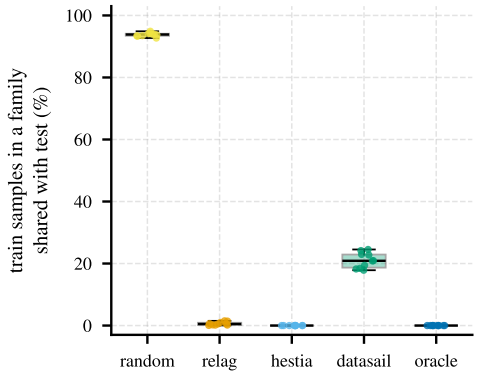
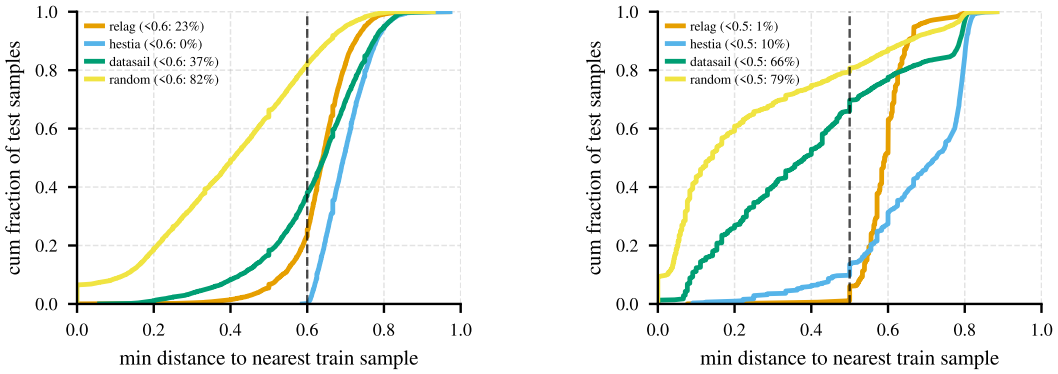


Figure 13: **Family leakage on the synthetic dataset.** Fraction of training samples whose family also appears in the test subset. Random: 93.8%, ReLaG: 0.62%, DataSAIL: 21.01%, Hestia: 0.00%, Oracle: 0.00%



(a) Small-molecule datasets

(b) Protein-sequence datasets

Figure 14: **Train–test distance across data-splitting methods.** ECDFs of the minimum distance from each test sample to its nearest training sample, pooled within modality: **(a)** six small-molecule datasets and **(b)** two protein-sequence datasets. Distance is $1-$ similarity, using Tanimoto similarity for molecules and global sequence identity (Needleman & Wunsch, 1970) for proteins. Dashed lines indicate the splitting thresholds (0.60 and 0.50, respectively); the legend reports the fraction of test samples closer to any train samples than the corresponding threshold.

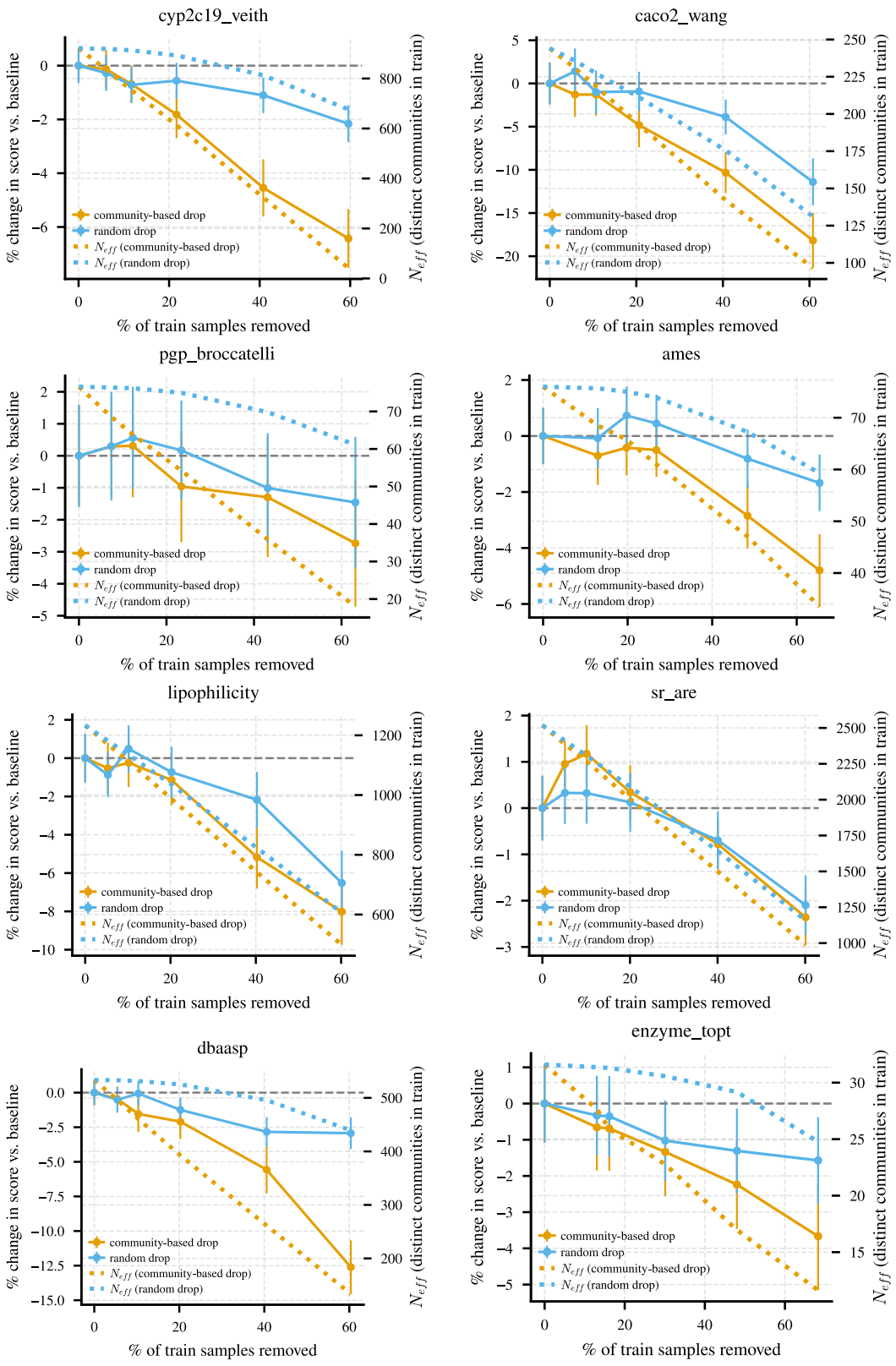


Figure 15: Effect on test performance of dropping random samples or full communities on different datasets. N_{eff} corresponds to the number of communities remaining in the dataset.