Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Self-Supervised Pretraining of Molecular Graph Encoders with LeJEPA

Michał Kulczykowski

michal.kulczykowski@deepsense.ai

deepsense.ai

Rafał Łabędzki

SGH Warsaw School of Economics

Research Article

Keywords:

Posted Date: October 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11151541/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Self-Supervised Pretraining of Molecular Graph Encoders with LeJEPA

Michał Kulczykowski*[1] and Rafał Łabędzki†[1,2]

[1]deepsense.ai
[2]SGH Warsaw School of Economics

Abstract

Self-supervised pretraining has transformed language and vision, but its value for molecular graph neural networks remains contested; we ask whether pretraining on a large unlabelled corpus improves downstream molecular property prediction. To answer it we adapt LeJEPA - a predictor-free joint-embedding predictive architecture whose embeddings are regularised by Sketched Isotropic Gaussian Regularisation (SIGReg) - to molecular graphs, and evaluate it with two encoder backbones (a GPS graph transformer and a Chemprop-style D-MPNN) on the Wong et al. [1] antibiotic-activity dataset and the `ogbg-molhiv` benchmark under a rigorous multi-seed, bootstrap-based protocol. We find that the benefit of pretraining is real at the level of the learned representation but does not robustly survive finetuning. A frozen probe on the pretrained embedding far exceeds the same architecture at random initialisation on both tasks (on `ogbg-molhiv`, ROC-AUC 0.788 vs 0.665, a +0.123 lift reaching the published self-supervised band), yet this edge does not convert into a downstream finetuning gain. On the antibiotic scaffold split it is significant on a single canonical partition (ΔAUPRC $= +0.041$, $p = 0.010$) but vanishes when replicated across five partitions (pooled +0.013, $p = 0.095$), and finetuning is null on the random split, on `ogbg-molhiv`, and with the D-MPNN backbone. The representational edge is nonetheless real and recoverable by other means. A width-controlled analysis shows the embedding saturates at $\sim$16–32 effective dimensions while the fingerprint keeps improving to 1024 bits, and that at matched dimensionality which representation wins depends on the split: the fingerprint leads on validation (0.799 vs 0.782 at 128 dimensions) and trails on the shifted test scaffolds (0.759 vs 0.788), which the embedding crosses with essentially no loss against the fingerprint's 0.040. Truncating the embedding to its informative subspace and concatenating it with a 1024-bit Morgan fingerprint raises `ogbg-molhiv` test ROC-AUC from 0.803 to 0.830 ($\Delta = +0.027$, 95% CI $[+0.003, +0.054]$, $p = 0.014$), while an untrained encoder put through the same pipeline gains nothing ($\Delta = -0.003$), so the pretrained embedding carries information the fingerprint does not. We conclude that LeJEPA pretraining yields a measurably better molecular representation whose downstream value is realised by feature-level combination rather than by finetuning, under which it gives at most a weak and partition-dependent gain that a single favourable split can make look significant. We release all pretraining code, configurations, and checkpoints at `https://github.com/GlockPL/MolecularLeJEPA`.

1 Introduction

1.1 Pretraining for molecular graph neural networks

Graph neural networks (GNNs) have become the standard tool for learning over molecular structure, treating a molecule as a graph of atoms and bonds and producing a fixed-size embedding suitable for property prediction [2]. Their accuracy, however, is bounded by the amount of *labelled* data available for the target property, and in most areas of chemistry experimental labels are scarce and expensive, since a typical discovery campaign measures a single endpoint on a few thousand compounds, while the space of synthesisable, drug-like

*Corresponding author. E-mail: `michal.kulczykowski@deepsense.ai`. ORCID: 0000-0001-9739-0911.
†ORCID: 0000-0002-0239-4125.

molecules numbers in the hundreds of millions [3]. This mismatch has motivated *self-supervised pretraining*, which learns general-purpose molecular representations from large pools of *unlabelled* structures and then transfers them to data-poor downstream tasks, mirroring the now-dominant "pretrain-then-finetune" recipe of language and vision.

Three families of self-supervised objectives have been explored for molecular graphs. *Contrastive* methods pull together augmented views of the same molecule while pushing apart views of different molecules, but require carefully designed augmentations and large numbers of negative samples. *Generative* methods, most prominently masked auto-encoding, reconstruct masked atoms or bonds in input space [4]; reconstruction forces the model to capture fine input detail that is not always useful downstream. *Predictive* methods, including the node- and graph-level attribute and context prediction of Hu et al. [5], sit between the two. Across these families the empirical picture is mixed. Pretraining helps on some endpoints and is neutral or harmful on others, and a well-tuned supervised baseline over hand-crafted molecular descriptors is often hard to beat [6]. This makes a careful, mechanistic study of *when* and *why* a given pretraining objective transfers as valuable as a new objective itself.

1.2 World models and LeJEPA

A complementary line of motivation comes from *world models*, predictive models of an environment that an agent can use to anticipate the consequences of its actions and to reason and plan in an internal latent space, rather than by trial and error in the world itself [7, 8]. Joint Embedding Predictive Architectures (JEPAs) are a particularly attractive way to build such models. Instead of modelling every aspect of an observation, a JEPA encodes observations into a compact, low-dimensional latent space and, from the rest, predicts the *latent* representation of a held-out part - a future observation, or another view. By predicting in representation space it focuses on the features that are actually predictable while discarding incidental, unpredictable detail; this distinguishes it from generative models, which must reconstruct the input in full, including detail that is often irrelevant, and from contrastive methods, which rely on large numbers of negative samples whose required count grows with the embedding dimension (the curse of dimensionality) [7].

The central difficulty in training a JEPA is *representation collapse*, whereby the prediction objective is trivially satisfied by mapping every input to the same constant embedding. Established JEPAs prevent this with asymmetric machinery - a momentum (exponential-moving-average, EMA) teacher encoder, stop-gradients, and a deliberately weakened predictor network; the graph-domain instantiation Graph-JEPA [9] follows exactly this recipe, predicting the latent codes of masked subgraphs with an EMA target encoder. The Latent-Euclidean JEPA (LeJEPA) [10] removes all of it, replacing the teacher, stop-gradient and predictor with a single explicit regulariser - Sketched Isotropic Gaussian Regularisation (SIGReg) - that drives the embedding distribution toward an isotropic Gaussian. Collapse then becomes impossible by construction, since a collapsed distribution is not isotropic, and the objective reduces to a transparent sum of a prediction term and a distributional regulariser with no asymmetric components to tune. The same SIGReg recipe has recently been shown to train stable world models for control directly from pixels (LeWorldModel [8]), underscoring that it is a general, modality-spanning approach to predictive representation learning. LeJEPA itself has so far been demonstrated on image data - natural and domain-specific image classification, with its pretrained features additionally supporting emergent unsupervised video object segmentation - and, to our knowledge, has not previously been applied to molecular graphs. This work adapts it to the molecular domain and studies, with deliberately rigorous controls, what its pretrained representations do and do not buy on downstream chemistry tasks.

1.3 Antibiotics and antimicrobial resistance

Our primary application is antibiotic discovery. Antimicrobial resistance (AMR) is among the most pressing threats to public health. Resistant infections already cause hundreds of thousands of deaths annually and, on current trends, threaten to render large parts of the antibiotic arsenal ineffective [11–14]. Yet the antibiotic pipeline has languished for decades - the chemical scaffolds that make good antibiotics are rare, and classical screening of large libraries is slow and costly [15]. Machine learning has recently re-energised the field. Stokes et al. [16] used a graph model to identify halicin, a structurally novel antibiotic, and Wong et al. [1] screened tens of thousands of compounds for activity against *Staphylococcus aureus* alongside human-cell

cytotoxicity, releasing a benchmark of 39,312 compounds with four binary endpoints. Antibiotic activity is extremely imbalanced - only about 1% of screened compounds are active - so the task is exactly the low-label regime in which transferable, pretrained representations should, in principle, help most.

1.4 A standard benchmark: `ogbg-molhiv`

Because a single niche assay is a weak basis on which to judge a pretraining method, we additionally evaluate on `ogbg-molhiv`, a standard molecular property-prediction benchmark from the Open Graph Benchmark [17]. It contains ∼41,000 molecules labelled for inhibition of HIV replication, with a fixed, canonical Bemis-Murcko scaffold split and ROC-AUC as the evaluation metric. Being on-distribution for our unlabelled pretraining corpus and equipped with many published scratch-versus-pretrained baselines, it lets us place LeJEPA pretraining in the context of the wider graph self-supervised-learning literature.

2 Method

2.1 Overview

We pretrain a molecular graph encoder f_θ that maps a molecule $G = (V, E)$, with atoms as the vertices V and bonds as the edges E, to a K-dimensional embedding $\boldsymbol{z} = f_\theta(G) \in \mathbb{R}^K$. From each molecule we generate several *views* - lightly corrupted whole-molecule graphs and connected subgraphs - and train the encoder so that the embeddings of all views of a molecule are mutually predictive, while the overall distribution of embeddings is held close to an isotropic Gaussian. Following LeJEPA [10], this is achieved *without* a predictor network, a momentum teacher, or stop-gradients. The entire objective is a convex combination of a prediction loss $\mathcal{L}_{\text{pred}}$ and a distributional regulariser $\mathcal{L}_{\text{SIGReg}}$, weighted by a single scalar λ (we use $\lambda = 0.2$). We define the two terms in the following subsections and state the full objective in Eq. 5. They play complementary roles. $\mathcal{L}_{\text{pred}}$ makes the representation informative (views of the same molecule agree), and $\mathcal{L}_{\text{SIGReg}}$ makes it non-degenerate (the embedding distribution stays full-rank and well-spread), so that the objective cannot be satisfied by collapsing all molecules to a single point.

2.2 View generation

View design is the heart of the method. For each molecule we generate $V_g = 2$ *global views* and $V_l = 8$ *local views* (Figure 1).

Global views. A global view is the whole molecule with a light corruption that removes a small amount of information without changing its identity. The first global view applies *atom-feature masking*, zeroing the feature vectors of a random 15% of atoms. The second applies *edge dropout*, removing a random 10% of bonds from the graph. Because both views retain essentially the entire molecule, their embeddings are nearly identical and serve as a stable, near-whole-molecule reference.

Local views. A local view is a connected subgraph grown by a breadth-first search (BFS) from a random seed atom until it covers a target fraction `cover_frac` of the molecule's atoms (we use 0.6). Local views are the source of difficulty in the objective. Predicting the whole-molecule reference from a 60% fragment forces the encoder to learn how local substructure determines global molecular identity.

Partial auxiliary conditioning. In addition to the learned graph features, we make a fixed vector of 217 z-scored RDKit physico-chemical descriptors available to the encoder. Crucially, these descriptors are attached to the *global views only*; local views receive a zero descriptor vector. This asymmetry is deliberate. The descriptor vector is a molecule-level constant, identical across all views of a molecule; attaching it to *every* view creates a trivial solution in which the encoder ignores structure and matches views on the shared descriptor signal, so the prediction loss collapses to zero without learning anything structural. Restricting descriptors to the global views enriches the prediction *target* (the global-view centroid) while leaving the local views to be explained from structure alone, which preserves the part-to-whole learning signal.

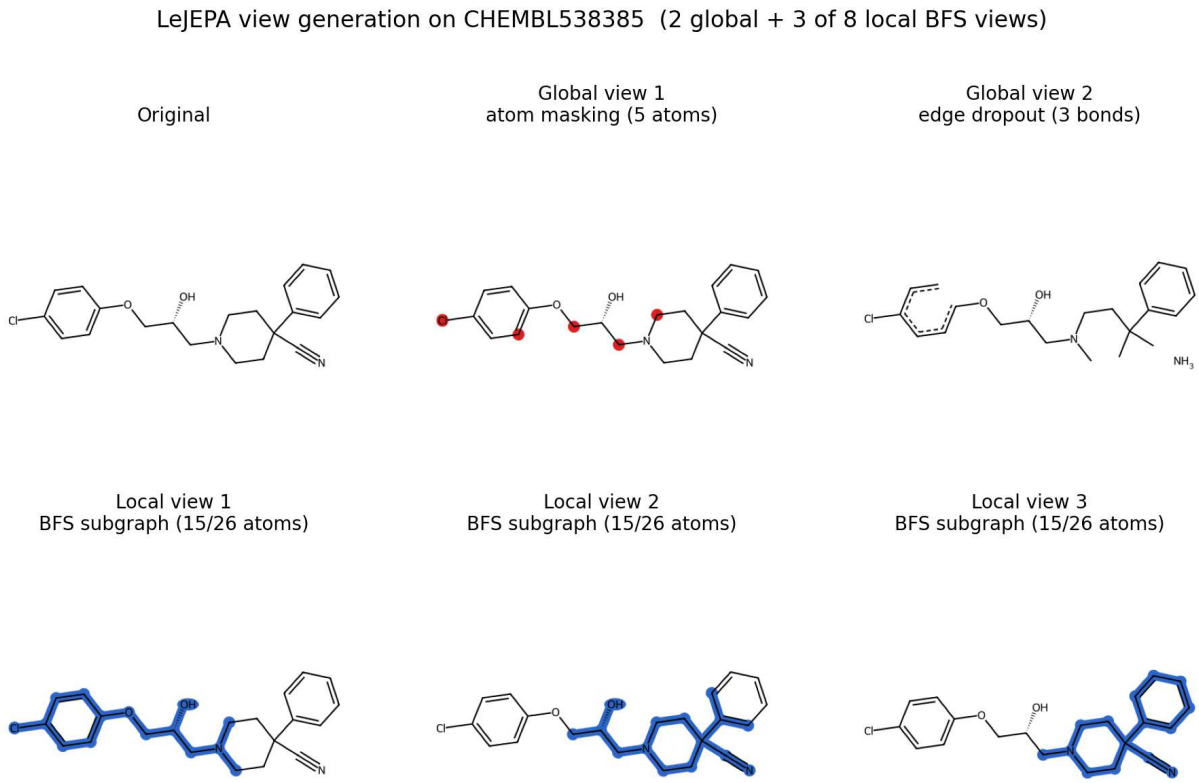


Figure 1: View generation for one molecule. Two global views (whole molecule with atom-feature masking and edge dropout) provide a near-whole-molecule reference; eight local views are connected BFS subgraphs covering ∼60% of the atoms. The encoder must predict the global reference from each local fragment.

2.3 Prediction loss

Let the $V = V_g + V_l$ views of a molecule be encoded to embeddings $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_V$, with the first V_g being the global views. We define the prediction target as the centroid of the *global-view* embeddings,

$$\boldsymbol{\mu} \;=\; \frac{1}{V_g} \sum_{v=1}^{V_g} \boldsymbol{z}_v, \tag{1}$$

and pull every view's embedding toward it,

$$\mathcal{L}_{\text{pred}} \;=\; \frac{1}{V} \sum_{v=1}^{V} \big\| \boldsymbol{\mu} - \boldsymbol{z}_v \big\|_2^2. \tag{2}$$

There is no separate predictor network and no stop-gradient; gradients flow through both $\boldsymbol{\mu}$ and each $\boldsymbol{z}_v$. Because the global views are near-whole-molecule and carry the auxiliary descriptors, $\boldsymbol{\mu}$ is a stable, informative target; the bulk of the learning signal therefore comes from the local-view terms, which require inferring the whole-molecule centroid from a fragment.

2.4 SIGReg loss

On its own, Eq. 2 is trivially minimised by mapping every molecule to the same point. SIGReg [10] prevents this by driving the batch of embeddings toward an isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ - a distribution that is full-rank, unit-variance in every direction, and uncorrelated across dimensions (Figure 2). It exploits a classical fact, that a distribution is an isotropic Gaussian if and only if *every* one-dimensional projection of it is a standard normal. SIGReg therefore "sketches" the embedding distribution by projecting it onto a large number of random unit directions ($N_s = 1024$) and, for each direction, measuring the deviation of the projected samples from $\mathcal{N}(0, 1)$ with a characteristic-function (Epps–Pulley) statistic. The regulariser is the average of these per-direction statistics (Figure 3). We adopt the SIGReg formulation of Balestriero et al. [10] unchanged. Concretely, for a batch of M embeddings $\{\boldsymbol{z}_n\}_{n=1}^{M}$ and a random unit direction $\boldsymbol{u}_j \in \mathbb{S}^{K-1}$, we compare the empirical characteristic function of the scalar projections $\boldsymbol{u}_j^\top \boldsymbol{z}_n$ with that of a standard normal, $\phi^*(t) = e^{-t^2/2}$, through the univariate Epps–Pulley statistic (Eqs. 3–4 restate their definitions)

$$\hat{\phi}_j(t) = \frac{1}{M} \sum_{n=1}^{M} e^{\,\mathrm{i}\, t\, \boldsymbol{u}_j^\top \boldsymbol{z}_n}, \qquad T(\boldsymbol{u}_j) = \int_{-\infty}^{\infty} \left|\hat{\phi}_j(t) - \phi^*(t)\right|^2 w(t)\, \mathrm{d}t, \tag{3}$$

with Gaussian window $w(t) = e^{-t^2/2}$ (which gives the integrand effectively compact support, so it is evaluated by the trapezoidal rule over $t \in [-5, 5]$). SIGReg averages this statistic over the $N_s = 1024$ random directions,

$$\mathcal{L}_{\text{SIGReg}} \;=\; \frac{1}{N_s} \sum_{j=1}^{N_s} T(\boldsymbol{u}_j). \tag{4}$$

By the Cramér–Wold theorem, matching every one-dimensional projection to $\mathcal{N}(0, 1)$ is equivalent to matching the full joint distribution to $\mathcal{N}(\mathbf{0}, \mathbf{I})$, so driving $\mathcal{L}_{\text{SIGReg}} \to 0$ makes the embedding isotropic Gaussian, which both prevents collapse and keeps the representation well-conditioned. The statistic is computed over the global batch under distributed training by averaging the per-direction empirical characteristic functions across replicas. We track the *effective rank* of the embedding covariance as a health metric; it rises toward K as SIGReg takes effect, and we use no learning-rate warmup - a warmup, in our experience, would let the prediction term collapse the representation before SIGReg engages. Putting the two terms together, the complete LeJEPA objective is

$$\mathcal{L} \;=\; (1 - \lambda)\, \mathcal{L}_{\text{pred}} \;+\; \lambda\, \mathcal{L}_{\text{SIGReg}}, \tag{5}$$

optimised end-to-end with $\lambda = 0.2$.

2.5 Encoder architectures

The objective in Eq. 5 is agnostic to the encoder; it requires only a map from a graph to a fixed-size embedding. We study two backbones.

GPS Graph Transformer. Our primary encoder is a GPS-style graph transformer [18] (Figure 4a). Each layer combines local message passing (a GINE convolution, which incorporates bond features) with a global multi-head self-attention block, so that every layer mixes short-range chemistry with long-range, whole-molecule context. Atoms are given a random-walk positional encoding (RWPE) [19] to break symmetries that message passing alone cannot. Node embeddings are mean-pooled to a graph embedding, optionally concatenated with the descriptor vector for global views, and projected to the K-dimensional output. We use a compact configuration (hidden dimension 128, 6 layers, 4 attention heads, $K = 128$; ∼2M parameters).

Chemprop-style D-MPNN. As an architecture control we also use a directed message-passing network (D-MPNN) in the style of Chemprop [20] (Figure 4b), reimplemented natively so that it consumes the same graph inputs and plugs into the same objective. Messages are passed along directed bonds (the message into an edge excludes its own reverse edge), and the final atom states are pooled to a graph embedding. We use a larger configuration here (hidden width 512, depth 5) to match the capacity at which this architecture is competitive on supervised tasks.

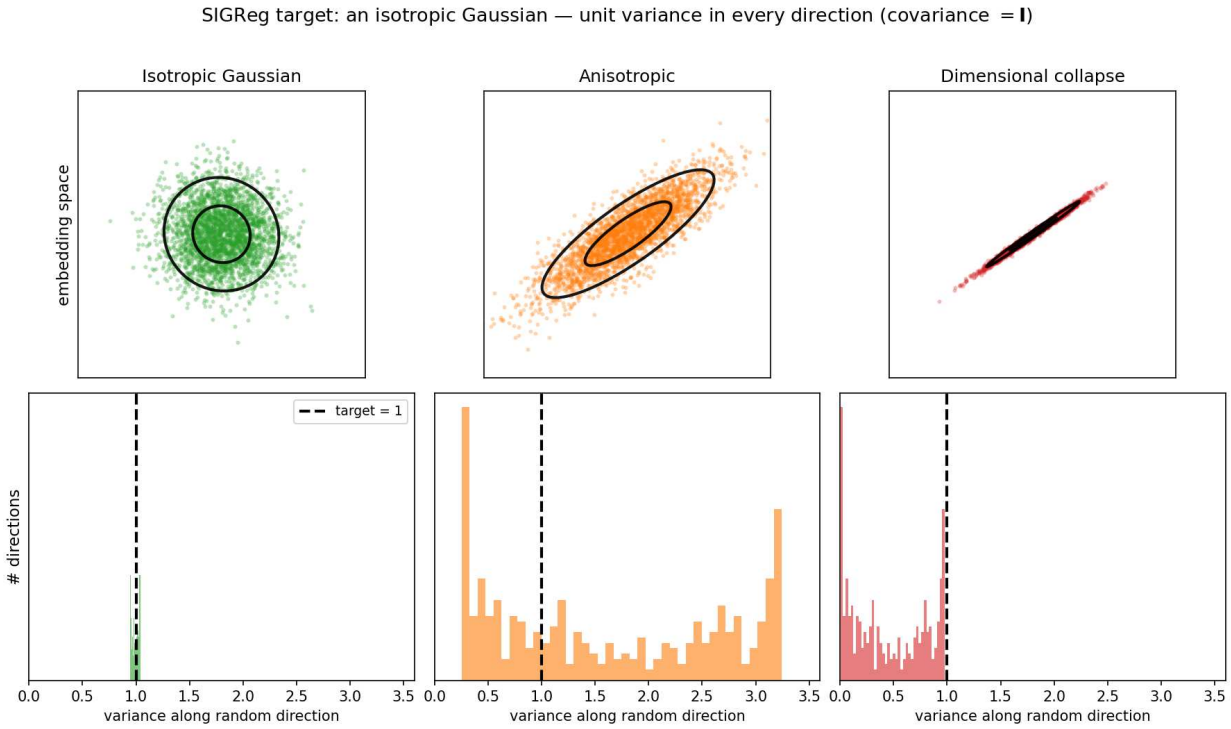


Figure 2: The SIGReg target. An isotropic Gaussian (left) has unit variance in every direction (covariance = $\mathbf{I}$); the failure modes SIGReg prevents are anisotropy (middle) and dimensional collapse (right). Bottom, the variance of the cloud along many random one-dimensional directions - isotropy means every such projection has unit variance.

2.6 Pretraining corpus and optimisation

Both encoders are pretrained on ∼2.9M bioactivity-curated molecules from ChEMBL. We use this corpus rather than a larger enumerated library because it covers substantially more of the downstream scaffolds relevant to our tasks. For `ogbg-molhiv`, which we can quantify exactly against its fixed split, ChEMBL contains 90.8% of the training scaffolds and 44.7% of the test scaffolds, and the median nearest-neighbour Tanimoto similarity between a test molecule and the corpus is 0.43. This coverage matches or exceeds that of the antibiotic task, so a null transfer result reflects the pretraining method rather than a corpus mismatch; the roughly 55% of test scaffolds absent from ChEMBL represent a genuine but unexceptional transfer ceiling. We optimise Eq. 5 with Adam, $\lambda = 0.2$, $V_g = 2$, $V_l = 8$, $N_s = 1024$ random projections, and no learning-rate warmup, using distributed data-parallel training across four GPUs. All pretraining code, configurations, and final checkpoints are released to support reproducibility; see the Code and data availability section.

2.7 Qualitative effect of pretraining

A first, qualitative check that pretraining learns chemically meaningful structure is shown in Figure 5, which projects the embeddings of ChEMBL molecules to two dimensions for the pretrained encoder and for a randomly-initialised encoder of the same architecture. To isolate the *learned* representation, both projections use the pure graph embedding with the auxiliary RDKit descriptors zeroed. Otherwise the descriptors, which directly encode these properties, would trivially organise either embedding. The pretrained embedding forms a well-spread manifold with smooth gradients in several physico-chemical properties (molecular weight, lipophilicity, ring count). The random-initialisation embedding recovers only a coarse molecular-size axis - a random message-passing network still encodes atom count, which correlates with molecular weight - but lacks the organised lipophilicity and ring-count gradients of the pretrained manifold, consistent with SIGReg

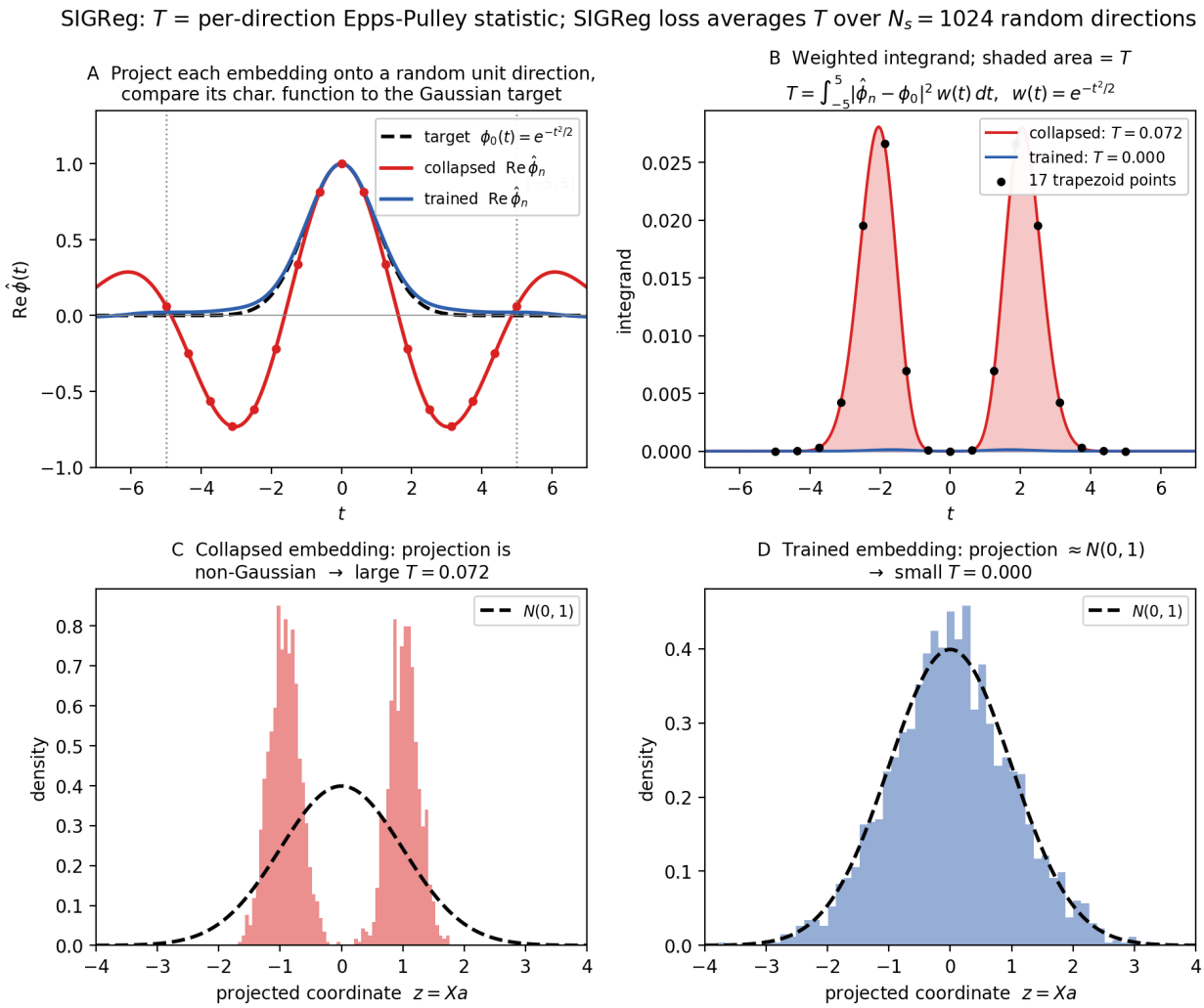


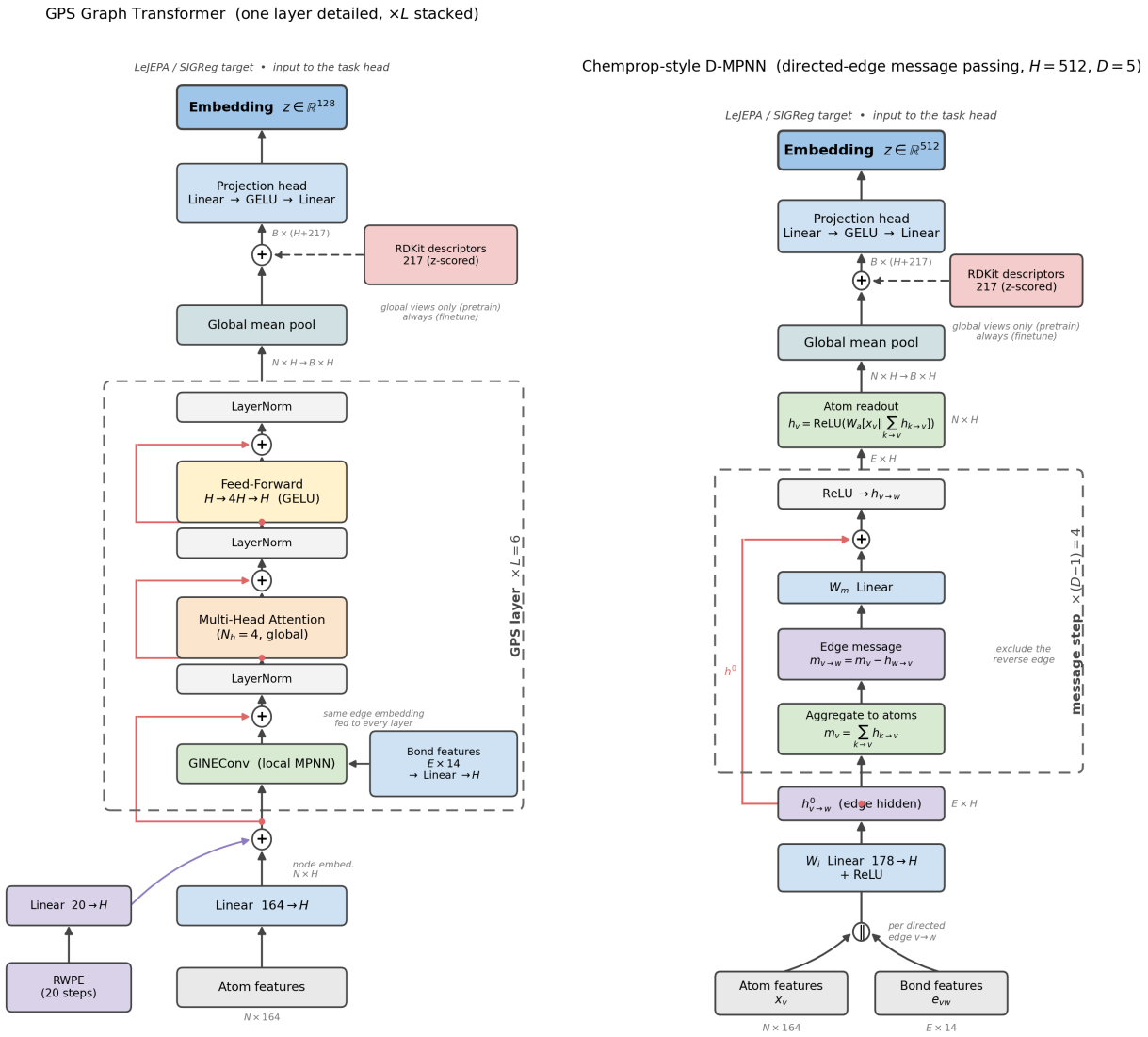
Figure 3: SIGReg in detail. Each embedding batch is projected onto random unit directions; per direction, the empirical characteristic function of the projection is compared to that of $\mathcal{N}(0, 1)$ via an Epps–Pulley statistic T (averaged over $N_s = 1024$ directions). A collapsed embedding yields non-Gaussian projections and large T; a well-trained embedding yields standard-normal projections and $T \to 0$.

having spread the representation into an isotropic, chemistry-organised distribution.

2.8 Downstream tasks and evaluation protocol

We evaluate the pretrained encoders on two downstream tasks, a niche antibiotic screen and a standard public benchmark, under the protocols below. Throughout, the same pretrained checkpoint is compared against the *same* architecture trained from scratch, so that the pretraining weights are the only difference between the two arms.

Antibiotic activity. The Wong et al. [1] dataset comprises 39,312 compounds assayed against *Staphylococcus aureus*, of which 512 (1.3%) are active. We use two splits. The *random* 80/20 split (stratified by label) reproduces the protocol behind the headline result of Wong et al. [1]. The *scaffold* split (Bemis-Murcko, with no scaffold shared between train and test) is the harder, more realistic test of generalisation to novel chemotypes. Because a scaffold split is only one of many leakage-free partitions of the same compounds, we do not rely on a single one: alongside the canonical Bemis-Murcko partition we generate four further



(a) GPS graph transformer.

(b) Chemprop-style D-MPNN.

Figure 4: The two encoder backbones, sharing an identical input featurization, descriptor pipeline, projection head, and training objective so that only the message-passing core differs. **(a)** The GPS graph transformer, in which each layer fuses a GINE local message-passing branch with a global self-attention branch; random-walk positional encodings are added at the input, and node embeddings are pooled to the molecule embedding. **(b)** The Chemprop-style D-MPNN, in which messages are passed along directed bonds (each edge message excludes its own reverse edge) and the final atom states are pooled. In both, descriptors are concatenated for global views before the projection head.

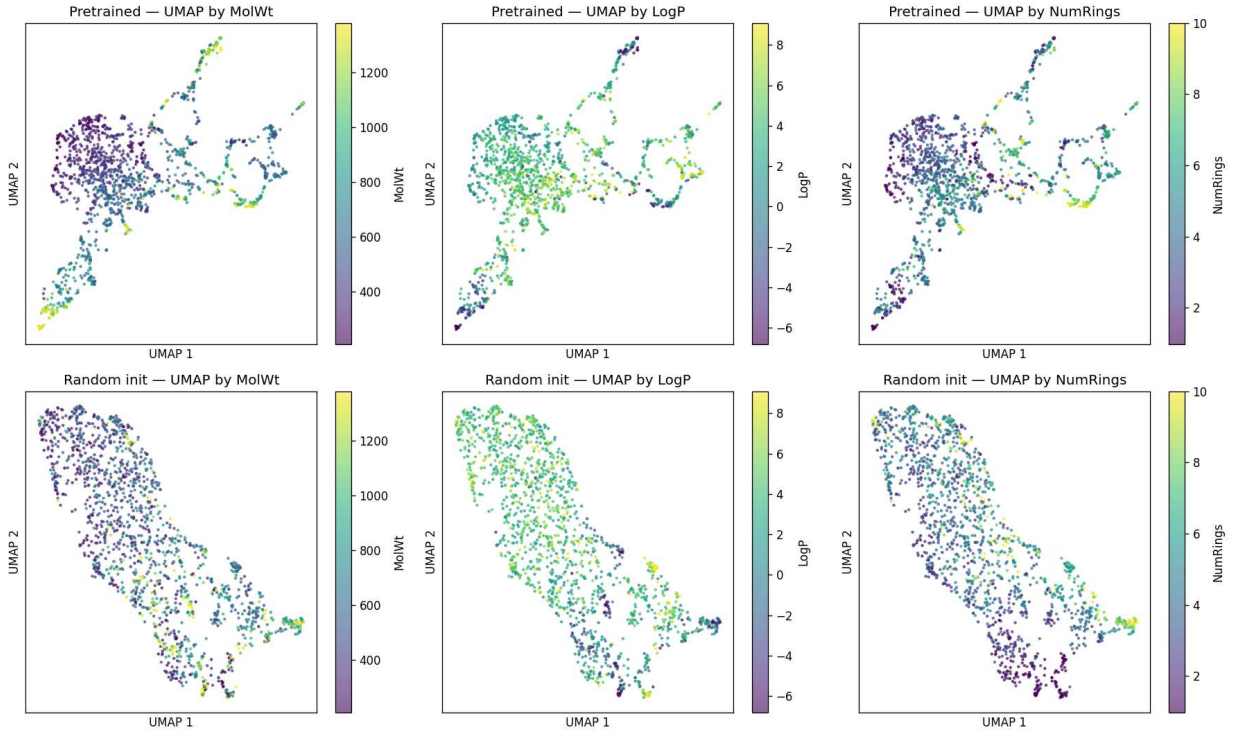


Figure 5: Two-dimensional projection of molecular embeddings, coloured by molecular weight, lipophilicity (LogP) and ring count, for the LeJEPA-pretrained encoder (top) versus a random-initialised encoder of the same architecture (bottom). Both use the pure graph embedding with the auxiliary descriptors zeroed. Pretraining produces a well-spread manifold organised by all three properties; the random encoder recovers only a coarse molecular-size (molecular-weight) axis.

leakage-free scaffold partitions and repeat the entire pretrained-versus-scratch comparison on each. The metric is the area under the precision–recall curve (AUPRC), appropriate for the extreme class imbalance. Each model is an ensemble of five seeds; for the random split, predictions are pooled over five independent splits (510 pooled positives) and reported as the pooled AUPRC with a 95% bootstrap confidence interval. Head-to-head model comparisons use a paired bootstrap over the pooled test predictions (10,000 resamples), which removes the variance shared by two models scored on the same molecules, and we quote one-sided p-values. We compare against the Chemprop result published by Wong et al. [1] (AUPRC 0.364 on their random split) and against our own Chemprop retrained on the identical splits.

`ogbg-molhiv`. A single niche assay is a weak basis on which to judge a pretraining method, so we repeat the scratch-versus-pretrained comparison on `ogbg-molhiv` [17], a benchmark of ∼41,000 molecules with a fixed Bemis-Murcko scaffold split and ROC-AUC as the metric. We follow the OGB protocol exactly, selecting the epoch on the validation set and reporting the corresponding test ROC-AUC averaged over five seeds. Because OGB does not prescribe the learning rate or the number of frozen-probe epochs, we sweep both and select by validation, tuning the from-scratch and pretrained models independently so that neither is disadvantaged.

Frozen probes. To separate the quality of the learned representation from what finetuning can recover on its own, we additionally evaluate each encoder frozen: weights are fixed, each molecule is mean-pooled to its 128-dimensional embedding, and a simple head is fit on those features with no gradient ever reaching the encoder. On the antibiotic scaffold split the head is an L^2-regularised logistic regression, which measures how much signal is encoded *linearly*. On `ogbg-molhiv` the head is a random forest, the strong tabular head that defines the top of the OGB leaderboard, reported as the mean ± s.d. over ten forest seeds on a fixed encoder.

In both cases the reference points are the same architecture at random initialisation and the corresponding hand-crafted representation (217 RDKit descriptors for the antibiotic task, a 2048-bit Morgan fingerprint for `molhiv`).

Width control and feature-level combination. A frozen probe that compares a 128-dimensional dense embedding against a 2048-bit sparse fingerprint confounds the quality of a representation with its dimensionality, because a random forest with `max_features` $= \sqrt{d}$ samples $\sim$45 of 2048 fingerprint columns per split but only $\sim$11 of 128 embedding columns. We therefore also sweep both representations against *retained dimensions* under the identical forest: the fingerprint over 64 to 4096 bits, and the embedding truncated two ways - a PCA top-k fitted on the training split (the best k-dimensional linear summary) and a random subset of k raw columns (no rotation, since a PCA rotation alone is not free on this probe, costing 0.011 on validation and 0.029 on test at full width). To test whether the two representations are complementary we then concatenate them, sweeping fingerprint width, retained embedding dimensions k, and `max_features` (which directly controls how often the embedding block is sampled at all), fitting the PCA on train only and selecting the reported cell on the validation split. For cost, these sweeps use smaller forests than the 2000-tree, ten-seed probe of Table 3 (500 trees and five seeds for the width sweep, 1000 trees and ten seeds for the concatenation sweep), which reads a few thousandths low in absolute terms; absolute values are therefore compared only within a sweep, while head-to-head differences use the same paired bootstrap over the shared test predictions (10,000 resamples) as elsewhere.

3 Results

3.1 Descriptor tree-ensembles match the graph model and exceed the published baseline

We first establish a strong, simple baseline of tree ensembles over the 217 RDKit molecular descriptors, with no graph network and no pretraining. Table 1 reports random-split test AUPRC. A linear model over the same descriptors reaches only 0.16, but nonlinear tree ensembles are far stronger - HistGBM 0.37, XGBoost 0.43, and a RandomForest 0.45 - confirming that the antibiotic signal carried by the descriptors is strongly nonlinear. The RandomForest ensemble **exceeds the published Chemprop result of** 0.364 **by** $\sim$0.09, using none of the graph machinery.

We compare against a Chemprop model we retrained on the identical splits (pooled AUPRC 0.407) with a paired bootstrap over the five pooled splits (510 positives), which removes the variance shared by two models scored on the same molecules. The RandomForest ensemble *significantly* outperforms this Chemprop (ΔAUPRC $= +0.046$, 95% CI $[+0.010, +0.081]$, one-sided $p = 0.006$); the gradient-boosted XGBoost ensemble sits a step lower, a statistical tie with Chemprop ($\Delta = +0.025$, $p = 0.086$). We therefore conclude that **a descriptor-only tree ensemble matches or exceeds a graph-based Chemprop and beats the published baseline** - the graph network is not necessary for this task. The same conclusion holds on the harder scaffold split, where the RandomForest ensemble reaches AUPRC 0.339 (95% CI $[0.269, 0.416]$), statistically level with the retrained Chemprop 0.312 (paired $\Delta = +0.023$, $p = 0.17$) and above the weaker HistGBM 0.281 and the linear floor 0.112. These descriptor baselines set a stringent reference against which any pretraining gain must be judged.

3.2 A frozen probe shows that pretraining learns a better representation

We next ask what pretraining does to the representation itself, before any finetuning is allowed to intervene, by freezing the encoder and fitting a simple head on the extracted embeddings. On both tasks the pretrained representation is clearly better than the same architecture at random initialisation.

On the antibiotic scaffold split (Table 2) the pretrained graph embedding alone reaches AUPRC 0.159 under an L^2 logistic regression, well above a randomly-initialised encoder of the same architecture (0.082) and above the 217 descriptors alone (0.112) - direct evidence that LeJEPA *does* learn antibiotic-relevant structure that random initialisation lacks, and that it is linearly accessible.

The same pattern holds, more strongly, on `ogbg-molhiv`. Freezing the encoder, mean-pooling each molecule to its 128-dimensional embedding, and fitting a random forest on top - the strong tabular head

Table 1: Antibiotic activity, *random* split, test AUPRC. Every model is evaluated by pooling its predictions over the same five random splits (510 pooled positives) and reported as the pooled point AUPRC with a 95% bootstrap confidence interval in brackets (10 000 replicates, identical resampling for all rows); the published value is quoted from Wong et al. [1] A paired bootstrap finds the RandomForest ensemble a significant win over retrained Chemprop ($\Delta = +0.046$, $p = 0.006$), XGBoost a tie ($\Delta = +0.025$, $p = 0.086$), and GPS+LeJEPA a tie ($\Delta = -0.001$).

Model (random split)	Test AUPRC
Logistic regression (descriptors)	0.164 [0.141, 0.194]
HistGBM ensemble (descriptors)	0.374 [0.331, 0.421]
XGBoost ensemble (descriptors)	0.432 [0.386, 0.478]
RandomForest ensemble (descriptors)	**0.453** [**0.407**, **0.500**]
Chemprop, retrained (graph)	0.407 [0.361, 0.457]
GPS + LeJEPA (graph)	0.406 [0.360, 0.456]
Wong et al. [1], published Chemprop	0.364

Table 2: Frozen linear probe (antibiotic, scaffold split), test AUPRC of an L^2 logistic regression fit on frozen features. The LeJEPA-pretrained graph embedding linearly encodes more antibiotic signal than a random-initialised encoder or the descriptors alone. Whether this representational edge survives full finetuning is the subject of Section 3.3.

Frozen features (linear probe)	Test AUPRC
Random-init graph embedding	0.082
RDKit descriptors	0.112
LeJEPA-pretrained graph embedding	**0.159**

that defines the top of the OGB leaderboard, with no gradient ever reaching the encoder - lifts the LeJEPA embedding to a test ROC-AUC of 0.788 ± 0.003, far above the *same* architecture at random initialisation (0.665 ± 0.004; Table 3). This +0.123 lift is attributable purely to the pretrained weights and places the embedding within the 0.75–0.79 band of published self-supervised GNNs on this benchmark. The same forest applied to a 2048-bit Morgan fingerprint scores 0.807 ± 0.004 - our faithful reproduction of the leaderboard's Morgan random-forest entry (ROC-AUC 0.806) - so the learned representation is competitive with, though it does not surpass, hand-engineered circular fingerprints.

Tracking the frozen probe across pretraining checkpoints shows that this representation forms early and does not depend on descriptor conditioning (Figure 6). The probe ROC-AUC climbs off the random-initialisation floor within the first ∼9 epochs and then plateaus near 0.79 for the remaining thirty; the ±0.003 epoch-to-epoch fluctuation on the plateau is random-forest seed noise, so individual per-epoch maxima carry no meaning and validation-based checkpoint selection lands anywhere inside the same plateau. Repeating the entire pretraining run with the descriptor vector removed ($n_{\text{desc}} = 0$, all other settings identical) reproduces this curve almost exactly, with a plateau-mean test ROC-AUC of 0.785 versus 0.787, identical maxima (0.794), and error bars that overlap at every epoch. The descriptors therefore add no descriptor-specific signal to the pooled graph representation the probe reads; they serve only the structural role of keeping the global-view prediction target non-trivial, and the pretraining gain is intrinsic to the graph encoder.

3.3 The representational gain does not robustly convert into a finetuning gain

A better frozen representation need not translate into a better finetuned model, and across four controlled comparisons - two evaluation regimes on the antibiotic task, a second backbone, and `ogbg-molhiv` - it largely does not.

Table 3: `ogbg-molhiv` frozen probe (scaffold split, test ROC-AUC; mean ± s.d. over ten random-forest seeds on a *fixed* encoder, so the spread reflects forest-seed variance, not encoder variance). The encoder is frozen and a random forest is fit on the pooled embedding, with no gradient reaching the encoder. LeJEPA pretraining lifts the embedding far above a random-initialised encoder of the same architecture and into the published self-supervised band, but does not surpass a Morgan fingerprint fed to the same forest.

Frozen features + random forest	Test ROC-AUC
Random-init GPS embedding	0.665 ± 0.004
GPS + LeJEPA embedding	$\mathbf{0.788 \pm 0.003}$
Morgan fingerprint (ECFP4, 2048 bits)	0.807 ± 0.004

Antibiotic, random split: null. A GPS encoder pretrained with LeJEPA and finetuned as an ensemble reaches a pooled AUPRC of 0.406, statistically indistinguishable from the retrained Chemprop (0.407; paired bootstrap $\Delta = -0.001$) and from the descriptor ensembles (Table 1). Under independent and identically distributed (i.i.d.) evaluation, pretraining neither helps nor hurts: supervised finetuning on an i.i.d. training set recovers the same signal on its own.

Antibiotic, scaffold split: significant on one partition, not across five. On the harder scaffold split, where the test set consists of chemotypes held out from training, a single canonical Bemis-Murcko partition does show a gain. Finetuning the same GPS encoder from the LeJEPA checkpoint rather than from scratch, across five seeds, raises test AUPRC from 0.232 ± 0.009 (scratch) to 0.271 ± 0.015 (pretrained) - an improvement in every one of the five seeds - with a paired bootstrap ΔAUPRC $= +0.041$ (95% CI $[+0.007, +0.079]$, one-sided $p = 0.010$).

This single-partition result is *not robust*. A scaffold split is only one of many leakage-free partitions of the same compounds, and a method that genuinely helps on novel chemotypes should help across them. Repeating the five-seed pretrained-versus-scratch comparison on four further scaffold partitions (Table 4) shows the effect is strongly partition-dependent. It is significantly positive on two partitions (the canonical one and partition 4), non-significant on one, and *negative* on the remaining two, with per-partition ΔAUPRC ranging from -0.033 to $+0.054$. Pooling all five partitions gives ΔAUPRC $= +0.013$ (95% CI $[-0.006, +0.032]$, one-sided $p = 0.095$), and treating each partition's Δ as a single observation gives a mean of $+0.010 \pm 0.038$ ($t = 0.60$); neither reaches significance. The apparent canonical-split gain is therefore largely an artefact of a favourable partition rather than a reliable benefit of pretraining.

A second backbone: null. The same comparison on a Chemprop-style D-MPNN backbone is null as well. Pretraining leaves it essentially unchanged, with a five-seed ensemble test AUPRC of 0.277 (pretrained) versus 0.278 (scratch) - ΔAUPRC $= -0.000$ (95% CI $[-0.045, +0.046]$, one-sided $p = 0.48$), only 2 of the 5 seeds improving - so even the fragile, partition-dependent GPS effect does not carry across architectures.

`ogbg-molhiv`: null. Finetuning the pretrained encoder does not improve on training from scratch on the public benchmark either (Table 5). The LeJEPA-pretrained encoder reaches a test ROC-AUC of 0.722 ± 0.010 against 0.717 ± 0.014 from scratch, a difference of $+0.005$ that lies within one standard deviation. The null is robust to tuning; across the entire learning-rate by probe-epoch grid the scratch and pretrained configurations both cluster in the 0.72–0.74 band and overlap completely, with no setting favouring the pretrained initialisation. Nor does the gap open up in the low-label regime where pretraining is expected to help most. At the smallest label budgets (2–10% of the training set) the pretrained initialisation is in fact marginally *worse* than scratch (Δ ROC-AUC between -0.03 and -0.07), because `molhiv` ROC-AUC already saturates near 0.72 at 2% of the data and leaves little headroom for a better initialisation to exploit. The antibiotic task tells the same low-label story: a few-shot analysis there shows the frozen pretrained representation does not surpass the descriptors when labels are scarce.

Taken together, these four comparisons indicate that **LeJEPA pretraining yields a measurably better molecular representation - evident under a frozen probe on two independent tasks (Section 3.2) - but this advantage does not robustly convert into a downstream finetuning**

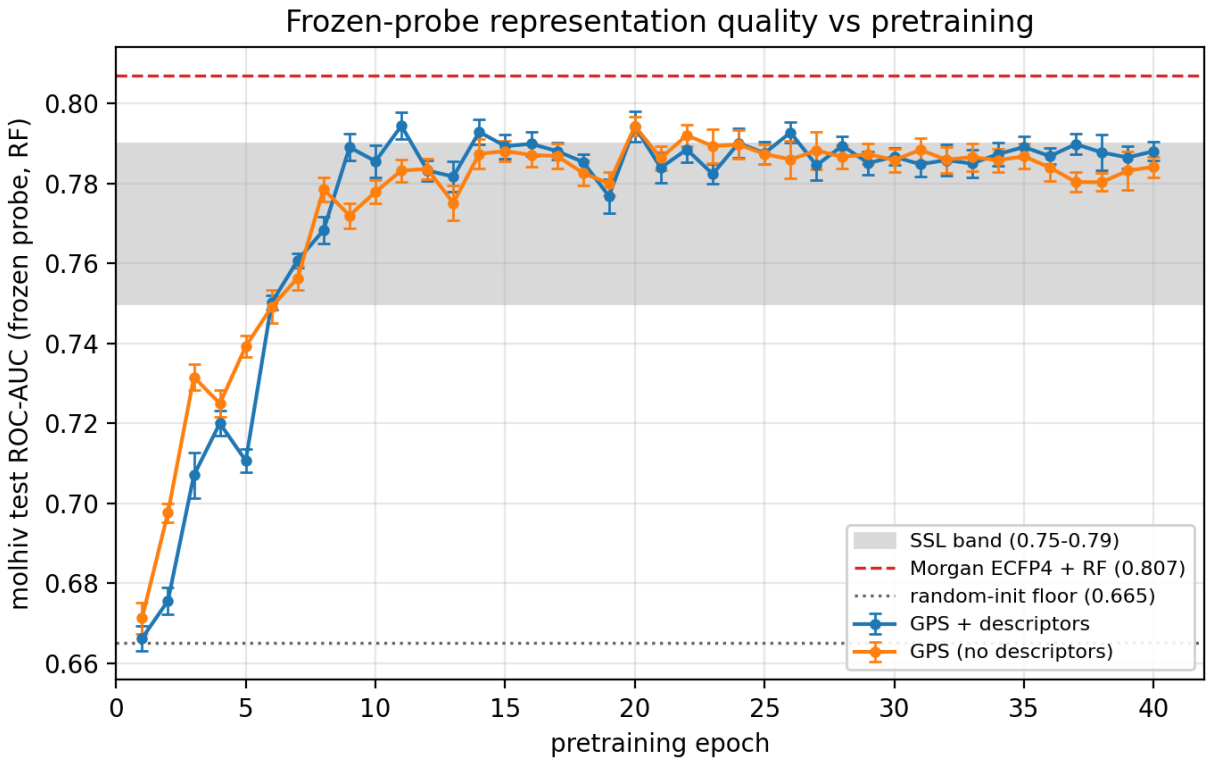


Figure 6: Frozen-probe representation quality across pretraining on `ogbg-molhiv`. At each pretraining epoch the encoder is frozen and a random forest (the leaderboard head) is fit on the pooled 128-dimensional embedding; markers are the per-epoch test ROC-AUC and error bars are ±1 s.d. over ten forest seeds. Both the descriptor-conditioned run (blue) and a descriptor-free run (orange; identical settings, $n_{\text{desc}} = 0$) rise off the random-initialisation floor within ∼9 epochs and then plateau just below the Morgan-fingerprint random forest, inside the published self-supervised band. The two curves are indistinguishable within seed noise, showing that the learned representation forms early and does not depend on descriptor conditioning.

gain. The apparent scaffold-split improvement does not survive multi-partition replication, and finetuning is null on the random split, on `molhiv`, and with a D-MPNN backbone.

3.4 The representation is complementary to a fingerprint

The finetuning nulls admit two readings. Either the pretrained representation is genuinely redundant with what supervised learning and hand-crafted features already supply, or our probes have been comparing it on unfavourable terms. The second reading deserves a test, because every frozen probe reported so far pits a 128-dimensional dense embedding against a 2048-bit sparse fingerprint under a random forest, a head that thrives on many weak, decorrelated columns: with `max_features` $= \sqrt{d}$ the forest samples ∼45 of the 2048 fingerprint bits at each split but only ∼11 of the 128 embedding dimensions. Dimensionality and representation quality are confounded.

The two representations differ in dimension-efficiency. Sweeping both against retained width under an identical forest separates them (Figure 7a). On validation the fingerprint improves steadily with width, from 0.747 at 64 bits to 0.839 at 1024, where it saturates (0.840 at 2048) and then declines (0.833 at 4096) as its own columns begin to dilute. The embedding does not improve beyond its first few dimensions: random subsets of raw columns reach 0.779 by 16 dimensions and are flat thereafter (0.775, 0.780, 0.776, 0.782 at 32, 64, 96 and 128), and a PCA truncation is likewise flat from 16 onward. The test curves agree for

Table 4: Antibiotic scaffold split, GPS + LeJEPA pretrained versus from scratch, across five leakage-free Bemis-Murcko scaffold partitions (no scaffold shared between train and test within each partition). Each arm is a five-seed ensemble; ΔAUPRC and the one-sided p come from a paired bootstrap over the pooled test predictions (10 000 resamples). The gain is significant on two partitions, reversed on two, and the pooled estimate is not significant - the single-partition result does not replicate.

Scaffold partition	Scratch	Pretrained	ΔAUPRC	p
canonical	0.232	0.271	+0.041	0.010
partition 1	0.310	0.277	−0.025	0.855
partition 2	0.313	0.304	+0.015	0.259
partition 3	0.247	0.223	−0.033	0.911
partition 4	0.289	0.336	+0.054	0.004
pooled	0.272	0.284	**+0.013**	**0.095**

Table 5: `ogbg-molhiv`, scaffold split, test ROC-AUC (OGB protocol, select on validation, report test; mean ± s.d. over five seeds, each at its independently validation-selected configuration). LeJEPA pretraining gives no significant improvement over training from scratch ($\Delta = +0.005$, within one s.d.).

Model (`ogbg-molhiv`, scaffold)	Test ROC-AUC
GPS, from scratch	0.717 ± 0.014
GPS + LeJEPA	0.722 ± 0.010

the fingerprint and for the raw-column embedding. They differ in one place: the PCA truncation, flat on validation, peaks near 32 on test and then falls to 0.759 as the uninformative tail is restored. The tail is thus not inert but actively harmful under the shift, which is the same dilution effect the concatenation experiment below runs into.

The first consequence is that the encoder does not fill the 128 dimensions it already has, so representational width is not the binding constraint and a wider backbone is not indicated; an untrained-backbone control agrees, gaining little from width (0.661, 0.691, 0.687 at hidden dimension 128, 256, 512).

The second is more informative, and cuts against the simpler claim that one representation is better than the other. At *matched* dimensionality the ranking *inverts between splits*: at 128 dimensions the fingerprint leads on validation (0.799 against 0.782) and trails on test (0.759 against 0.788). The reason is that the two degrade very differently across the scaffold shift - the fingerprint loses 0.040 ROC-AUC from validation to test, while the embedding loses nothing at all (it scores 0.006 *higher* on the shifted split). The fingerprint's absolute advantage comes entirely from continuing to improve in a width regime where the embedding has already saturated. So the two representations differ not only in how much information they extract, but in how much of it survives a change of chemotype - the same asymmetry the next experiment exploits, visible here without any combination at all.

Truncated, the embedding adds to the fingerprint. This also explains why a naive concatenation fails. Appending all 128 embedding dimensions to a 2048-bit fingerprint yields 2176 columns of which the forest samples ∼47 per split, so ∼2.8 are embedding dimensions and, since only ∼32 of the 128 carry signal, well under one *informative* embedding column is available at a typical split, while the remaining dimensions supply noise to overfit. Using each representation near its own optimum instead - the fingerprint at 1024 bits, the embedding truncated to a PCA top-k fitted on the training split - changes the outcome (Table 6). Selecting the cell with the best *validation* ROC-AUC, over fingerprint width, k and `max_features`, and likewise selecting the fingerprint-only baseline on validation, gives 1024 bits plus a 32-dimensional truncation, whose test ROC-AUC is 0.830 ± 0.004 against the baseline's 0.803 ± 0.004, each the mean over the forest seeds. A paired bootstrap over the shared test predictions, averaged across those seeds, puts the difference at $\Delta = +0.027$ (95% CI $[+0.003, +0.054]$, one-sided $p = 0.014$); measured instead against the strongest fingerprint-only cell on test rather than on validation, it is +0.025 (95% CI $[+0.001, +0.051]$, $p = 0.019$).

Table 6: `ogbg-molhiv` frozen probe, fingerprint and pretrained embedding combined (random forest, 1000 trees, mean ± s.d. over ten forest seeds, `max_features` = 0.1). The PCA truncation is fitted on the training split only. Configurations are selected on *validation*: doing so picks the 1024-bit fingerprint for the baseline arm and the 1024-bit fingerprint plus a 32-dimensional truncation for the combined arm (bold). Note that the combined representation is *not* better on validation; it is better on the shifted test scaffolds, and the $k = 64$ row is the dilution control in which the gain disappears. The random-init rows repeat the sweep with an *untrained* backbone of the same architecture under the same truncation, and show that graph-derived columns as such do not help a fingerprint: only pretrained ones do.

Frozen features + random forest	dim	Valid ROC-AUC	Test ROC-AUC
Each representation alone			
LeJEPA embedding, PCA-32	32	0.769	0.789 ± 0.004
Random-init embedding, PCA-32	32	0.754	0.681 ± 0.003
Morgan fingerprint, 1024 *bits, plus a truncated embedding*			
Fingerprint alone	1024	0.845	0.803 ± 0.004
+ LeJEPA PCA-16	1040	0.841	0.827 ± 0.003
+ LeJEPA PCA-32	1056	**0.847**	$\mathbf{0.830 \pm 0.004}$
+ LeJEPA PCA-64 (dilution control)	1088	0.831	0.811 ± 0.003
+ random-init PCA-32 (control)	1056	0.836	0.800 ± 0.004
+ random-init PCA-64 (control)	1088	0.842	0.791 ± 0.004
Morgan fingerprint, 512 *bits*			
Fingerprint alone	512	0.821	0.790 ± 0.005
+ LeJEPA PCA-32	544	0.815	0.821 ± 0.002
+ random-init PCA-32 (control)	544	0.821	0.773 ± 0.004

Restoring the diluting dimensions removes the effect: at $k = 64$ the difference falls to +0.008 (95% CI $[-0.021, +0.039]$, $p = 0.29$), a control that identifies dilution rather than the extra columns as the mechanism.

The gain requires pretraining, not merely extra graph-derived columns. A fingerprint gains nothing from a dense block of columns per se, so the comparison above could in principle reflect the *form* of the added features rather than what pretraining put in them. Repeating the entire sweep with an untrained backbone of the same architecture, the same PCA truncation fitted the same way, and the same selection rule settles this (Table 6). The untrained embedding is a far weaker representation on its own (0.681 against the pretrained 0.789), and appending it to a fingerprint does not help: in the configuration the pretrained embedding wins on, the untrained one gives $\Delta = -0.003$ (95% CI $[-0.029, +0.026]$, $p = 0.59$), and selecting the untrained arm's own best cell on validation gives $\Delta = -0.013$ (95% CI $[-0.045, +0.021]$, $p = 0.77$). Across all twelve cells the pretrained embedding beats its untrained twin in every one, and the untrained columns are net harmful in most, reaching −0.030 at 512 bits. The complementarity is therefore a property of what LeJEPA learned, not of the dimensionality, density or provenance of the added features.

The gain is robustness to scaffold shift, not extra fit. Where this improvement lives is more informative than its size, and qualifies it. On the validation scaffolds the concatenation is not better than the fingerprint alone: 0.847 against 0.845, a difference of +0.002 that is well within noise (Figure 7b). The two arms separate only on the test scaffolds, because they degrade differently across the split: the fingerprint alone loses 0.042 ROC-AUC from validation to test, the concatenation loses 0.017, and the diluted $k = 64$ control loses an intermediate 0.020. The embedding is therefore not adding signal that the validation chemotypes reward; it is adding signal that survives the change of chemotype. This is consistent with the frozen-probe evidence that pretraining organises the representation by physico-chemical structure rather than by task-specific detail, and it is the mechanism one would want from a pretrained representation. It should nonetheless be read with the caveat it implies: because the gain is invisible on validation, it cannot be selected *for*, and a single shifted test set with 130 actives is a narrow base on which to measure it.

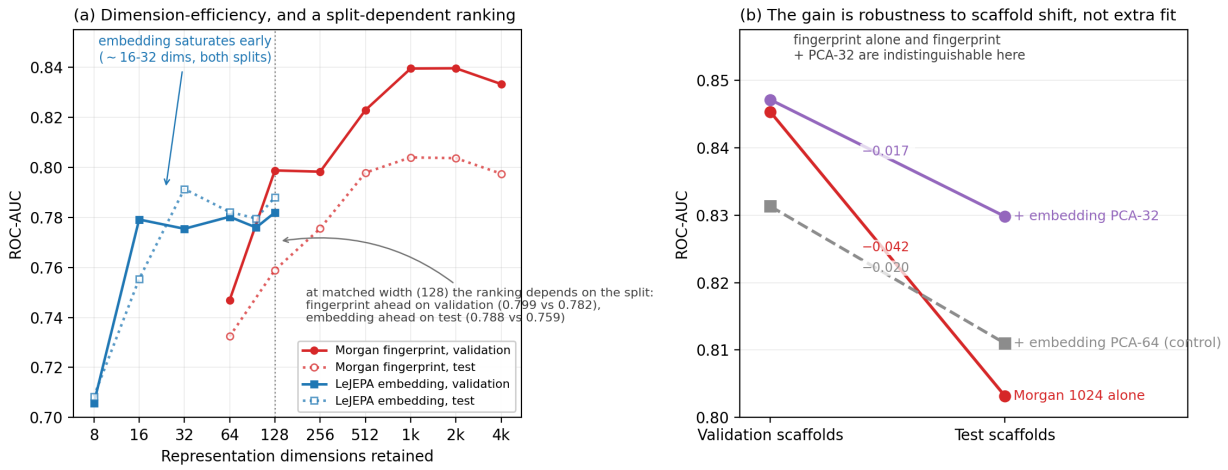


Figure 7: The pretrained embedding is complementary to a fingerprint once its informative subspace is isolated. **(a)** Frozen-probe ROC-AUC against retained dimensions under an identical random forest, on the validation (solid) and test (dotted) scaffolds. The fingerprint improves to ∼1024 bits; the embedding saturates by ∼16–32 dimensions. At the matched width of 128 the ranking inverts between the splits, the fingerprint leading on validation and the embedding on test, because the fingerprint loses 0.040 ROC-AUC across the shift while the embedding loses none. **(b)** Where the combination helps. On the validation scaffolds a 1024-bit fingerprint and the same fingerprint plus a 32-dimensional truncation of the embedding are indistinguishable; they separate on the test scaffolds because the combination degrades far less across the split (-0.017 versus -0.042). Restoring the diluting dimensions ($k = 64$) removes most of the protection.

3.5 Positioning against published baselines

Two comparisons place these results in context. First, our compact 128-dimensional backbone reaches only ∼0.72 on `molhiv` even from scratch, modestly *below* the 0.75–0.79 band occupied by published self-supervised GNNs on this benchmark (for example a GIN trained from scratch [21], MolCLR [6], and GraphMAE [4]). We therefore make no claim to a competitive absolute number; the informative quantity is the controlled internal scratch-versus-pretrained comparison, which is null. Second, the top of the OGB leaderboard [17] *corroborates* our antibiotic finding rather than contradicting it. Its strongest `molhiv` entry is a molecular-fingerprint plus random-forest ensemble (ROC-AUC ≈ 0.85) that incorporates graph networks only marginally and uses no pretraining, while the best graph-transformer entry (Graphormer [22], ≈ 0.81) owes its score to *supervised* quantum-property transfer rather than self-supervision. As on the antibiotic task, hand-engineered fingerprints remain the strongest *single* representation; but, also as on the antibiotic task, LeJEPA pretraining demonstrably learns useful structure, its frozen embedding far exceeding a random initialisation and reaching the competitive self-supervised band (Table 3) even though full finetuning renders that advantage redundant. The strongest representation we obtain is neither of the two alone but their combination (Section 3.4), whose 0.830 exceeds every single representation we measured and narrows the distance to that leaderboard-topping fingerprint ensemble - achieved by adding 32 columns to 1024, rather than by scaling either component.

4 Discussion and conclusion

We adapted LeJEPA, a predictor-free joint-embedding predictive architecture regularised by Sketched Isotropic Gaussian Regularisation (SIGReg), to molecular graphs, and evaluated whether the resulting self-supervised pretraining improves downstream property prediction. Our central finding is that the benefit of pretraining is real at the level of the learned *representation* - a frozen probe on the pretrained embedding clearly beats a randomly-initialised encoder on two independent tasks - but does not robustly carry through to a downstream *finetuning* gain, the one apparently-significant transfer result failing to replicate across scaffold partitions. That benefit is not, however, unusable: it can be realised at the feature level, by isolating the

embedding's informative subspace and combining it with a fingerprint.

Four results support this conclusion. First, on the Wong et al. [1] antibiotic dataset a tree ensemble over 217 RDKit descriptors - using no graph network and no pretraining - matches or beats a graph-based Chemprop and exceeds the published baseline, establishing that the graph model is not required for this task and setting a stringent reference for any pretraining gain. Second, LeJEPA pretraining nonetheless learns a representation that, frozen, linearly encodes more antibiotic-relevant signal than a randomly-initialised encoder (0.159 vs 0.082 AUPRC) and that on `ogbg-molhiv` reaches the published self-supervised band (0.788 vs 0.665 ROC-AUC under a frozen random-forest probe, a +0.123 lift) - a real and reproducible representational improvement. Third, this representational advantage does not robustly convert into a downstream finetuning gain. The one finetuning result that reaches significance - the antibiotic scaffold split, ΔAUPRC $= +0.041$ on a single canonical partition - does not replicate across five scaffold partitions, where it ranges from -0.033 to $+0.054$ and the pooled estimate is $+0.013$ (95% CI $[-0.006, +0.032]$, $p = 0.095$, not significant). Finetuning is likewise null under in-distribution (random-split) evaluation, on which supervised learning recovers an equivalent solution unaided, and on `ogbg-molhiv`. The effect is also architecture-dependent. The same pretraining gives a Chemprop-style D-MPNN backbone no measurable benefit even on the canonical scaffold split (ΔAUPRC $= -0.000$, 95% CI $[-0.045, +0.046]$, $p = 0.48$).

Fourth, and against that run of nulls, the representational advantage *does* convert downstream when it is combined with a fingerprint instead of being finetuned. A width-controlled analysis shows the embedding saturates at $\sim$16–32 effective dimensions, and that at matched dimensionality the ranking inverts between splits: the fingerprint leads on validation and the embedding on the shifted test scaffolds, which it crosses with essentially no loss against the fingerprint's 0.040. Truncating the embedding to that subspace and concatenating it with a 1024-bit fingerprint raises `ogbg-molhiv` test ROC-AUC from 0.803 to 0.830 ($\Delta = +0.027$, 95% CI $[+0.003, +0.054]$, $p = 0.014$), with a $k = 64$ dilution control in which the gain disappears. An untrained backbone put through the identical pipeline gains nothing ($\Delta = -0.003$, $p = 0.59$), so this is a consequence of pretraining rather than of adding dense graph-derived columns to a fingerprint.

The coherent reading is that LeJEPA pretraining supplies representational structure that is most useful precisely when the downstream task demands generalisation beyond the training distribution, and is otherwise redundant with what supervised learning and hand-engineered descriptors already provide. The concatenation result makes this concrete rather than merely interpretive: the combined representation is no better than the fingerprint on the validation scaffolds (0.847 vs 0.845) and better only on the test scaffolds, because it loses far less across the shift (-0.017 vs -0.042). What pretraining contributes is robustness to a change of chemotype, which is invisible to any metric computed on data drawn like the training set. This is consistent with the broader observation, reinforced by the descriptor and fingerprint baselines on both tasks, that low-order physico-chemical features carry much of the signal in these property-prediction problems.

Several limitations bound these conclusions. The downstream finetuning evidence is the weak point of the pretraining case. The one significant transfer result, on the canonical antibiotic scaffold partition, does not survive replication across five scaffold partitions (Table 4), so we report it as partition-dependent rather than as a reliable gain. The robust, reproducible signal is at the level of the representation (the frozen probes on both tasks), and why finetuning fails to exploit a representation that a random forest can exploit remains the central open problem this work leaves. The concatenation result carries its own caveats. It rests on one benchmark, one split and 130 test actives; the truncation dimension is a hyperparameter, chosen on validation, which is defensible but must be described as fitted; and because the benefit appears only under the scaffold shift and not on validation, it is a quantity that cannot be selected for in the ordinary way. Our compact encoder attains absolute scores below the strongest published self-supervised graph networks, so our claims concern the controlled scratch-versus-pretrained contrast rather than leaderboard standing. Finally, the pretraining corpus does not fully cover the downstream chemical space, which upper-bounds the achievable transfer. As to where effort is best directed, our width analysis argues *against* simply enlarging the encoder, since the present one does not fill the dimensions it has; the more promising directions are pretraining objectives or auxiliary signals that are complementary to - rather than redundant with - molecular descriptors, evaluation designed around distribution shift rather than in-distribution accuracy, and treating the pretrained embedding as a component to be combined with established representations rather than as a replacement for them.

Code availability

All code and configurations are available under the MIT licence, developed openly at `https://github.com/GlockPL/MolecularLeJEPA` and archived at `https://doi.org/10.5281/zenodo.22920159`. This comprises the LeJEPA implementation (SIGReg, the view generators and the prediction loss), both encoder architectures, the pretraining, finetuning and evaluation loops, every configuration file used to produce the reported runs, and the statistical-analysis scripts.

The `ogbg-molhiv` experiments of Section 3.4, including the fingerprint-embedding concatenation, its dilution control, its random-initialisation control and the paired bootstraps, are packaged as a self-contained reproduction with step-by-step instructions, the expected values of every cell, and the outputs of our own run for comparison. The saved ensemble predictions are included, so the significance tests can be re-executed in seconds without refitting any model.

Data availability

The data generated and analysed during this study are available as follows.

Data generated by this study are archived at `https://doi.org/10.5281/zenodo.22920159` and developed openly at `https://github.com/GlockPL/MolecularLeJEPA` under the MIT licence. The archive contains the three pretrained backbones - the GPS encoder, its descriptor-free ablation and the D-MPNN control - from which every result we report is derived; the exact deterministic train/validation/test partitions for the antibiotic task in both the random and the scaffold split, so that the splits themselves can be inspected rather than merely regenerated; and the per-compound test predictions of every model. All reported metrics, bootstrap confidence intervals and p-values are computed from those predictions, and therefore recompute from the archive alone.

Previously published data were analysed as follows. The pretraining corpus is ChEMBL release 37 (EMBL-EBI, `https://doi.org/10.6019/CHEMBL.database.37`), rebuilt from the public release by the download and featurisation scripts in the archive rather than redistributed. The antibiotic screening labels are the Supplementary Data of Wong et al. [1] (`https://doi.org/10.1038/s41586-023-06887-8`), redistributed in the archive with attribution to the original publication. The `ogbg-molhiv` benchmark is part of the Open Graph Benchmark [17], downloaded automatically by the `ogb` package, and its canonical scaffold split is used unmodified.

The finetuned model ensembles ($\sim$4.2 GB in total) are not deposited, because they are not required to verify any result reported here: every metric, confidence interval and p-value recomputes from the per-compound predictions included in the archive. They are available from the corresponding author on request, and are needed only to regenerate predictions from weights rather than to reuse the committed ones.

Declarations

Funding

No funding was received for conducting this study.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Ethics approval

Not applicable. This study involved no human participants, no human data or tissue, and no animals. It uses only previously published, publicly available chemical datasets, as detailed in the data availability statement above.

Consent to participate and consent for publication

Not applicable, for the reasons given under Ethics approval.

Author contributions

M.K. conceived the study, designed and implemented the pretraining and evaluation pipelines, performed all experiments and statistical analyses, prepared the figures, and wrote the manuscript. R.Ł. critically reviewed the manuscript and the analyses, and contributed to the interpretation of the results and to the presentation of the work. Both authors read and approved the final manuscript and agree to be accountable for all aspects of the work.

Declaration of generative AI use

The authors used a generative AI assistant (Anthropic's Claude, through the Claude Code command-line interface) throughout this work. Its use covered implementation and analysis - writing and refactoring parts of the pretraining, finetuning, evaluation and plotting code, and helping to build the statistical analysis scripts - and the preparation of this manuscript, where it assisted with drafting, editing and LaTeX formatting. It was not used to generate experimental data. All research questions, experimental designs and interpretations are the authors' own; every training and evaluation run reported here was launched and supervised by the authors, and every number, table and figure was checked against the underlying experimental output. The authors reviewed and edited all AI-assisted code and text, and take full responsibility for the content of this article.

References

(1) Wong, F. et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature* **2024**, *626*, 177–185, DOI: 10.1038/s41586-023-06887-8.

(2) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. In *International conference on machine learning*, 2017, pp 1263–1272.

(3) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *Journal of Chemical Information and Modeling* **2020**, *60*, 6065–6073, DOI: 10.1021/acs.jcim.0c00675.

(4) Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; Tang, J. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp 594–604.

(5) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* **2019**.

(6) Wang, Y.; Wang, J.; Cao, Z.; Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence* **2022**, *4*, 279–287.

(7) LeCun, Y. et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review* **2022**, *62*, 1–62.

(8) Maes, L.; Le Lidec, Q.; Scieur, D.; LeCun, Y.; Balestriero, R. LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels. *arXiv preprint arXiv:2603.19312* **2026**.

(9) Skenderi, G.; Li, H.; Tang, J.; Cristani, M. Graph-level Representation Learning with Joint-Embedding Predictive Architectures. *arXiv preprint arXiv:2309.16014* **2023**.

(10) Balestriero, R.; LeCun, Y. LeJEPA: Provable and Scalable Self-Supervised Learning Without the Heuristics, 2025.

(11) O'neill, J. Antimicrobial resistance: tackling a crisis for the health and wealth of nations. *Rev. Antimicrob. Resist.* **2014**.

(12) Tang, K. W. K.; Millar, B. C.; Moore, J. E. Antimicrobial Resistance (AMR). *British Journal of Biomedical Science* **2023**, *Volume 80 - 2023*, DOI: 10.3389/bjbs.2023.11387.

(13) Aslam, B.; Asghar, R.; Muzammil, S.; Shafique, M.; Siddique, A. B.; Khurshid, M.; Ijaz, M.; Rasool, M. H.; Chaudhry, T. H.; Aamir, A.; Baloch, Z. AMR and Sustainable Development Goals: at a crossroads. *Globalization and Health* **2024**, *20*, 73, DOI: 10.1186/s12992-024-01046-8.

(14) Amann, S.; Neef, K.; Kohl, S. Antimicrobial resistance (AMR). *European Journal of Hospital Pharmacy* **2019**, *26*, 175–177, DOI: 10.1136/ejhpharm-2018-001820.

(15) Walsh, C. Where will new antibiotics come from? *Nature Reviews Microbiology* **2003**, *1*, 65–70.

(16) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackermann, Z., et al. A deep learning approach to antibiotic discovery. *Cell* **2020**, *180*, 688–702.

(17) Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

(18) Rampášek, L.; Galkin, M.; Dwivedi, V. P.; Luu, A. T.; Wolf, G.; Beaini, D. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* **2022**, *35*, 14501–14515.

(19) Dwivedi, V. P.; Luu, A. T.; Laurent, T.; Bengio, Y.; Bresson, X. In *International Conference on Learning Representations (ICLR)*, 2022.

(20) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M., et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling* **2019**, *59*, 3370–3388.

(21) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. In *International Conference on Learning Representations (ICLR)*, 2019.

(22) Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.