LAVA: EXPLAINABILITY FOR UNSUPERVISED LATENT EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised black-box models are drivers of scientific discovery, but are difficult to interpret as their output is often a multidimensional embedding rather than a well-defined target. While explainability for supervised learning estimates how input determines model output, its unsupervised counterpart should uncover how input is captured by the learned latent structure. However, existing adaptations of supervised model explainability for unsupervised learning deliver explanations either at the single-sample or the dataset-summary level, too fine-grained or reductive to be meaningful, and cannot explain models without a mapping function. To address this, we propose LAVA, a model-agnostic method to explain embedding structure through feature covariation in the input data. Using prototype-like explanations, LAVA captures local patterns of input feature correlations that may reoccur globally across the embeddings. LAVA offers stable explanations at a chosen granularity, revealing domain-relevant patterns such as pixel motifs in images and disease signals in cellular processes otherwise missed by existing methods.

1 INTRODUCTION

An explanation can be defined as an answer to a "why" question (Fraassen, 1988). Traditional models such as Newton's law of gravitation, and statistical models such as linear regression, are inherently explanations as they allow reasoning about observable phenomena through rigorous mathematical description. Nevertheless, for more complex problems, e.g., facial recognition or protein structure prediction, expert-led modeling is superseded by machine learning (ML), where flexible model architectures and automated end-to-end optimization algorithms take center stage. ML has had considerable success in leveraging high-dimensional data for scientific discovery across fields such as chemistry, mathematics, and biology (Jumper et al., 2021; Davies et al., 2021; Movva et al., 2019; Ching et al., 2018). However, even as ML models make reliable predictions or accurately reflect real-world phenomena, interpreting their behavior is notoriously difficult because most models are "black boxes", with complexity of the data and operations required to produce an output exceeding what is practically comprehensible (Lipton, 2017; Humphreys, 2009; Beisbart & Räz, 2022). Limited insight into model behavior is a barrier to adoption of ML for decision making, especially in critical areas such as healthcare (Rudin, 2019; Murdoch et al., 2019; Ching et al., 2018; Wang et al., 2023a). To improve understanding, the community relies on two key strategies: intrinsically interpretable but restrictive models, or explainability for post-hoc interpretation of black-box models (Rudin, 2019; Murdoch et al., 2019; Molnar, 2022a).

The increasing use of black-box ML poses new challenges to existing explainability methods. At its core, post-hoc explainability aims to enable reasoning about the output of a model in terms of its input, which also allows to cross-reference model behavior with domain knowledge for verification, reliability assessment, and bias mitigation. Existing explainability methods cover a wide range of approaches, including feature or sample importance, anchors, prototypes, and counterfactuals (Molnar, 2022b). However, most methods are developed for supervised ML, whose output is usually a well-defined target. This overlooks unsupervised black-box ML, which are designed to reveal latent structure with limited prior knowledge and relevant for knowledge discovery in many fields. Indeed, embeddings based on autoencoder and transformer architectures have shown effectiveness in language, image, and molecular biology tasks (Brown et al., 2020; Ramesh et al., 2021; Makrodimitris et al., 2024), and serve as a basis of foundation models (Bommasani et al., 2022). Likewise, manifold learning is prominent in bioinformatics for data exploration, trajectory inference, cell-type or

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.

spatial domain identification, and multiomics integration (Kobak & Berens, 2019; Brbić et al., 2022; Saelens et al., 2019; Wolf et al., 2019; Haviv et al., 2024). Improving our understanding of these learned latent structures could thus contribute meaningfully to contemporary scientific discovery.

Arguably, explainability for unsupervised ML shifts focus away from the "why" question of interpreting a decision, which requires an a priori known target of interest, towards the more exploratory "what" question of understanding which patterns in the data are captured by the model or the learned latent structure (Watson, 2023; Miller, 2019). Crucially, most unsupervised black-box models output multi-dimensional embeddings. However, relating latent axes to the input data is not straightforward, as they do not correspond to labeled quantities and are therefore uninterpretable (Crabbé & Schaar, 2022b). Research on disentanglement further indicates that, without inductive biases or supervision, latent dimensions remain entangled and cannot be meaningfully explained in isolation (Locatello et al., 2020; Crabbé & Schaar, 2022b). As such, a promising direction for this type of explainability could be linking the relative, multivariate organization of the embeddings to the underlying data. Additionally, since many manifold learning algorithms produce embeddings without a parametrized mapping function (Murphy, 2022), explainability methods should also be able to leverage information about the latent spatial relations exclusively from the embeddings themselves.

Literature on explainability for unsupervised latent embeddings is scarce. Notably, Crabbé & Schaar (2022b) propose a label-free explainability (LFXAI) framework, adapting linear feature and sample importance to the unsupervised setting. While the approach sensibly re-purposes existing tools, it exhibits several limitations. First, it aggregates explanations obtained separately per latent dimension, forgoing the relative spatial organization contained in the multivariate embedding representation. Second, it yields single-sample or dataset-summary explanations that are generally too fine-grained or reductive to describe the latent structure. Third, since it relies on embeddings with a mapping function, it cannot be applied to popular non-parametric manifold learning methods like t-SNE and UMAP (Meilă & Zhang, 2024; Maaten & Hinton, 2008; McInnes et al., 2020). While parametric versions of these methods exist (Sainburg et al., 2021; Damrich et al., 2023; Nazari et al., 2023), they do not address the broader representation issue. Nor do studies into hyperparameter effects (Becht et al., 2019; Kobak & Berens, 2019; Wang et al., 2021; Xia et al., 2024) and neighborhood retention (Han et al., 2022), which assess properties of embedding methods but offer limited understanding of the structure they uncover. Similarly, targeted explainability for manifold learning tends to focus on single-sample or whole-dataset feature attribution, rather than latent structure, and is model-specific (Lambert et al., 2022; Couplet et al., 2023). Finally, more manual approaches relate embedding organization to input data, but rely on prior knowledge or predefined features of interest, and are infeasible for high-dimensional data (Liu et al., 2019; Bibal et al., 2021).

We argue that explaining black-box embeddings requires new strategies for revealing what patterns in the data are captured by a learned multivariate latent structure. To that end, we propose Locality-Aware Variable Associations (LAVA): a post-hoc, model-agnostic method to explain latent embeddings by relating their relative spatial organization to variation in the input data. LAVA generates prototype-like explanations that capture local patterns of input feature correlation and reveal their reoccurrence throughout representative latent embedding neighborhoods (Fig. 1). Neighborhoods capture the potentially nonlinear latent organization at a chosen granularity, while feature correlations reflect dependencies present in the underlying data. We use LAVA to explain embeddings of the MNIST, bag-of-words, and single-cell kidney datasets, showcasing its ability to discover domain-relevant modules of pixel, word, and gene expression correlation. We further characterize properties of LAVA explanations, assess their stability, and compare against the LFXAI framework.

2 METHODOLOGY

LAVA creates explanations that relate local organization of latent embeddings to feature covariation in three steps: (1) locality definition, (2) locality representation, and (3) module extraction (Fig. 1). Step (1) defines the scale and reference points (probes) of neighborhoods, or localities, to capture latent organization. To determine probe locations, we design a custom weighted k-means formulation based on in-degree centrality. Step (2) creates a representation of each locality, linking it to the input data through feature correlations. Step (3) uses our proposed Association Matrix Factorization (AMF) to generate explanations as modules of correlation patterns reoccurring across localities.

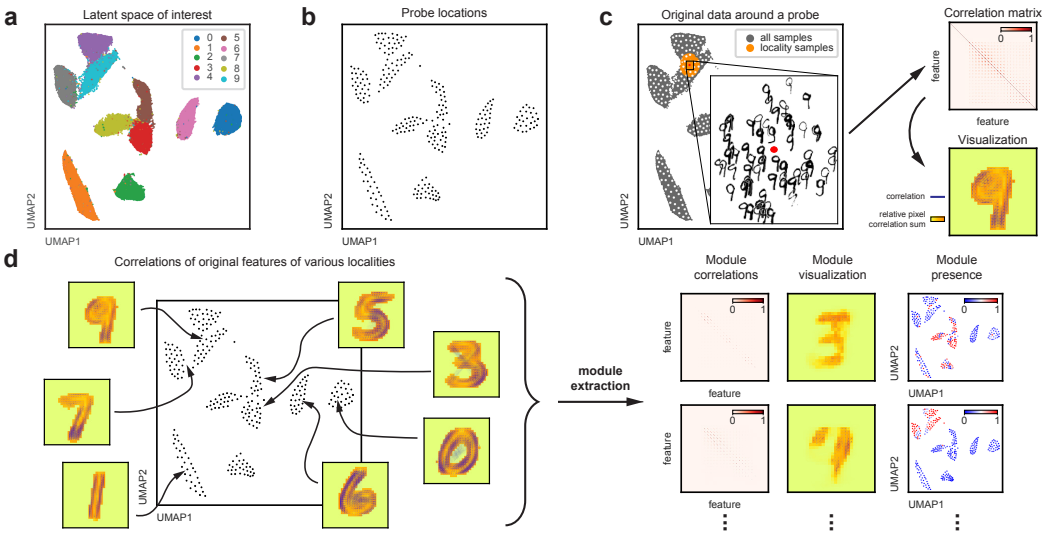


Figure 1: **LAVA. a**, LAVA explains embeddings of a dataset (e.g. UMAP embeddings of MNIST). **b**, Embeddings are captured via ℓ localities: overlapping n-sized neighborhoods centered at probes. **c**, Each locality is represented by feature-to-feature correlations based on its original samples (top right, correlation matrix; bottom right, heatmap of pixelwise correlation sums, lines link correlated pixels). **d**, LAVA extracts explanations as modules: correlation patterns reoccurring across localities.

2.1 Locality Definition

To reflect local organization of embeddings, LAVA uses a smaller number of localities, centered around probes (not samples) spread across the latent space (Fig. 1b). A locality centered at probe p is an n-sized Euclidean neighborhood containing the n samples nearest to p in the latent space. The number of localities $\ell \leq E$ is given by $\ell = E \times o/n$, based on the number of embeddings E, neighborhood size n, and factor o controlling the extent of overlap between localities (App. A.1).

Locality placement. The goal is to determine the locations of the ℓ probes around which the localities will be centered. To ensure that localities are representative of the embeddings, LAVA optimizes locality in-degree centrality towards neighborhood in-degree centrality (Figs. S21, S27, and S34). In other words, the proportion of probe-centered localities including a given sample is made to approximate the proportion of all sample-centered n-neighborhoods including that same sample. Practically, LAVA finds the locations of the ℓ probes as cluster centers using a custom weighted k-means formulation, where $k = \ell$. The clustering uses the following sample-weighting scheme:

$$w_i = \left(\frac{in_neighborhood_i}{E}\right)^{\alpha} \times \left(\frac{1}{\bar{d}_i^n}\right)^{\beta}, \tag{1}$$

where w_i is the weight of sample i, $in_neighborhood_i$ is the in-degree centrality of sample i using sample-based n-neighborhoods, $\bar{d}_i^n$ is the average distance from sample i to its n nearest neighbors, and α and β are parameters optimized using the DIRECT algorithm (MacQueen, 1967; Jones et al., 1993). We define the cost function for DIRECT as the Manhattan distance between the neighborhood and locality in-degree centralities of each sample:

$$\mathcal{L}(\alpha, \beta) = \sum_{i=1}^{E} \left| \frac{in_neighborhood_i}{E} - \frac{in_locality_{i,\alpha,\beta}}{\ell} \right|, \tag{2}$$

where $in_locality_{i,\alpha,\beta}$ is the number of localities around the resulting ℓ cluster centers that include i as a neighbor. LAVA uses only the cluster centers, not the clusters themselves (App. A.1).

2.2 Locality Representation

After defining localities, LAVA links these units of latent organization to the original data. Different types of representation are possible. As proof-of-concept, we characterize the behavior of data

within a locality using feature covariation, aiming to capture feature dependencies as a hallmark of high-dimensional data, and focusing on patterns rather than values. Each locality is represented by a vector of $\binom{D}{2}$ pairwise correlations between the D features based on the locality's samples, using absolute Spearman correlation to capture monotonic association values in $[0, 1]$ (Fig 1c). Spearman correlation is more flexible than Pearson, as it can capture nonlinear relations, while still remaining computationally inexpensive and interpretable. Correlations are filtered out (set to 0) if one of the features has the same value in $> 75\%$ of the locality. Locality representations are collected in matrix $\boldsymbol{C}$, with rows and columns respectively denoting the ℓ localities and the $\binom{D}{2}$ feature pairs. Alternative representations are beyond the scope of this work, but briefly discussed in App. A.3.

2.3 Module Extraction

Finally, LAVA extracts explanations of the input embeddings as modules of feature correlation patterns reoccurring across localities. Conceptually, LAVA modules are prototype-like explanations (Molnar, 2022c), but are based on robust reoccurring patterns instead of representative samples. Consequently, LAVA modules have presence values, quantifying in which localities they occur. To extract these interpretable explanations, LAVA uses our proposed AMF method, which adheres to the following two properties: (1) multiple, possibly overlapping modules can be co-present in a locality; and (2) modules capture reoccurring rather than average patterns, since averages would not be faithful to correlations of individual localities. Together, these properties distinguish AMF from the more conventional Non-negative Matrix Factorization (NMF; Lee & Seung, 1999).

Briefly, AMF factorizes locality matrix $\boldsymbol{C}$ into module matrix $\boldsymbol{M}$ and presence matrix $\boldsymbol{P}$ (Fig. 2): $\boldsymbol{M}$, of size $M \times \binom{D}{2}$, contains the correlation pattern for every module over the feature pairs; $\boldsymbol{P}$, of size $\ell \times M$, yields the presence of each module in every locality. Entries are real values in $[0, 1]$. We ensure properties (1) and (2) through a specific form of factorization and loss function, respectively.

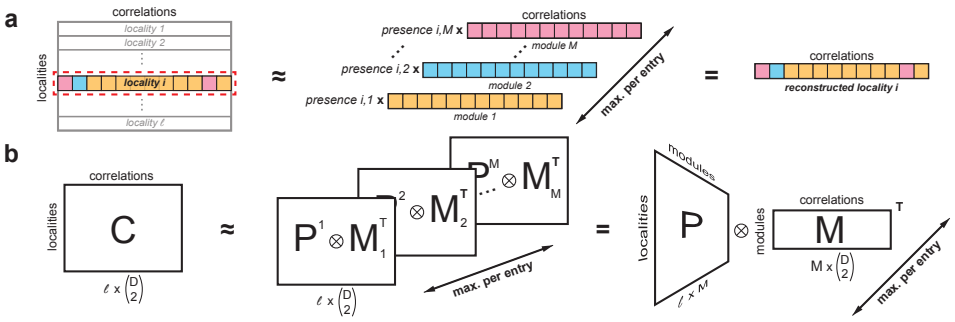


Figure 2: **AMF module extraction.** **a**, Locality i is reconstructed by multiplying each module by its presence, then taking entrywise maximum over M modules. **b**, Localities are denoted by matrix $\boldsymbol{C}$, with ℓ locality-based (rows) correlation vectors of $\binom{D}{2}$ feature pairs (columns). AMF approximates $\boldsymbol{C}$ based on modules $\boldsymbol{M}$ and presences $\boldsymbol{P}$ with a max-per-entry of the outer product between columns of $\boldsymbol{P}$ (superscript) and rows of $\boldsymbol{M}$, yielding a tensor of M matrix slices (same shape as $\boldsymbol{C}$). Reducing dimension M via a per-element maximum yields the reconstructed localities.

AMF factorization form. Since correlations are not additive, we avoid summing across modules and use maximums instead. This allows modules to contribute overlapping patterns of correlation, without using addition, to ensure property (1). LAVA reconstructs each correlation j of locality i by taking the maximum of the correlations across modules multiplied by their presences for locality i:

$$\hat{\boldsymbol{C}}_{ij} = \max\left(\boldsymbol{P}_i \odot (\boldsymbol{M}^T)_j\right), \tag{3}$$

where $\odot$ is the Hadamard product and single letter subscripts denote matrix rows. Reconstruction of locality i is then an element-wise maximum across products of modules and presences (Fig. 2a):

$$\hat{\boldsymbol{C}}_i = \max_{1 \le j \le \binom{D}{2}} \left(\boldsymbol{P}_{i1} \times \boldsymbol{M}_1^T, \boldsymbol{P}_{i2} \times \boldsymbol{M}_2^T, \ldots, \boldsymbol{P}_{iM} \times \boldsymbol{M}_M^T\right), \tag{4}$$

Across ℓ localities, we take element-wise maximums of M matrices (Fig. 2b). Intuitively, reconstruction is like matrix multiplication $\boldsymbol{P} \times \boldsymbol{M}$, with elements as maximums instead of product sums.

AMF loss function. The AMF loss contains two terms: a custom L2-weighted cosine distance term including an overestimation mask, designed to make modules capture dominant patterns of feature covariation, and thus ensure property (2); and a regularization term to keep scale consistent between modules and the original locality correlations. For locality i, the loss is:

$$\mathcal{L}(\boldsymbol{C}_i, \hat{\boldsymbol{C}}_i) = \frac{||\boldsymbol{C}_i||_2}{\sum_{j\neq i}^{\ell} ||\boldsymbol{C}_j||_2} \, d_{cos}\left(\mathbf{w}_i \odot \boldsymbol{C}_i, \mathbf{w}_i \odot \hat{\boldsymbol{C}}_i\right) + \gamma \, \text{MAE}(\boldsymbol{C}_i, \hat{\boldsymbol{C}}_i), \tag{5}$$

where $||\,||_2$ are $L2$ norms, $\hat{\boldsymbol{C}}_i$ is the reconstructed locality (Eq. 4), $\odot$ is the Hadamard product, $d_{\cos}$ is cosine distance $d_{\cos}(\mathbf{x}, \mathbf{y}) = (1 - \cos(\mathbf{x}, \mathbf{y}))$, and MAE is the scale regularization via mean absolute error, with a strength hyperparameter γ. The overestimation mask $\mathbf{w}_i$ is defined as:

$$w_{ic} = \begin{cases} \nu, & \text{if } \hat{\boldsymbol{C}}_{ic} > \boldsymbol{C}_{ic} \\ 1, & \text{otherwise} \end{cases}, \tag{6}$$

where w_{ic} is the weight for locality i and correlation c, and ν is an up-weighing hyperparameter. The rationale for the loss is as follows: cosine distance ensures we capture reoccurring patterns rather than absolute values; localities are weighted by their $L2$ norms to favor concentrated over spread-out patterns; and overestimation errors are penalized like in quantile regression (Koenker & Hallock, 2001). Finally, MAE regularization ensures modules the same scale as the locality correlations, despite the otherwise scale-invariant loss. Property (2) can then be assessed based on overestimation errors (Sec. 3.2). We optimize using ADAM (App. A.4; Kingma & Ba, 2017).

Number of modules M. We choose M based on reconstruction and stability, similar to an approach for NMF (Alexandrov et al., 2013). For stability, we find M-medoid clusters using all $R \times M$ modules and calculate silhouette width (Rousseeuw, 1987) based on cosine similarity. Higher stability indicates similar modules across runs. Lower M guides modules toward more globally reoccurring patterns. However, too low M might force local patterns to merge, causing poorer reconstruction. We look for an elbow of the loss over M, resolving ambiguity by favoring higher stability (Sec. 3.1).

2.4 Runtime

LAVA is scalable with parallelization ($\mathcal{O}$-notation and runtime analysis in App. B.11). Step 1, locality placement, computes n-sized nearest neighborhoods of E embeddings (dimensionality D_L), and then runs DIRECT k-means with $k = \ell$ multiple times to find probes for the ℓ localities, evaluating centrality. Runtime experiments show each parameter has an approximately linear dependence with time. Step 2, locality representation, calculates pairwise Spearman correlations between D features, and ℓ localities of size n. Ranking D features for n samples of each locality is $\mathcal{O}(\ell \; D \; n \; \log n)$, correlations of the ranked data are $\mathcal{O}\left(n \; D^2 \; \ell\right)$. Both are parallelized over ℓ and D, and thus scalable. Step 3, AMF module extraction, uses stochastic gradient optimization (ADAM) to find M modules in locality data of size $l \times \binom{D}{2}$, thus we omit $\mathcal{O}$-notation. Runtime experiments show an approximately quadratic dependence on D, linear on M, and sublinear on ℓ.

3 Experiments

We perform several experiments to verify and showcase the utility of LAVA explanations. First, we evaluate the stepwise and end-to-end stability of the LAVA method. Second, we investigate whether LAVA explanations adhere to the two sought-after properties. Third, we compare LAVA against the label-free explainability framework (Crabbé & Schaar, 2022b). Finally, we use LAVA to explain UMAP-embedded kidney single-cell gene expression data and reveal patterns related to disease.

Datasets. We use three datasets: MNIST images (Lecun et al., 1998), KPMP[1] single-cell kidney gene expression (Lake et al., 2023), and the "Bag-of-Words" (BOW) collection of text documents (Newman, 2008). The KPMP data measure activity of ∼30k genes (features) for ∼225k kidney cells (samples) (Stark et al., 2019); we remove mitochondrial and ribosomal genes, and use the top 1000 most variable genes following best practices (App. B.2; Heumos et al., 2023). The BOW data are a collection of news and scientific article documents. We use the top 1000 most variably occurring words, with their relative frequencies in documents as features for LAVA (App. B.9).

[1] Kidney Precision Medicine Project: https://www.kpmp.org/.

Embeddings. We embed MNIST with UMAP and default parameters (McInnes et al., 2020); or a convolutional autoencoder to compare with LFXAI (Crabbé & Schaar, 2022b). For KPMP, we use public UMAP embeddings. For BOW, we run TF-IDF with 300 components, then t-SNE with default parameters (App. B.9). We apply LAVA to each embedding set, with neighborhood $n = 3000$ for MNIST and $n = 500$ for KPMP / BOW, and overestimation weight $\nu = 9$ for all experiments.

3.1 Stability of LAVA Explanations

We assess stepwise and end-to-end stability of LAVA results. **Setup.** We apply LAVA to UMAP embeddings of MNIST and KPMP, and assess stability of localities and modules as follows. ***Localities***: we run locality placement and representation (steps 1-2) $R_L = 5$ times, for different values of overlap o; then calculate $\binom{D}{2}$ feature-pair average correlations across localities, and compare those between runs per o value using cosine similarity (Fig. 3a). ***Modules***: using a single locality placement run with $o = 12$ for MNIST and $o = 3$ for KPMP, whose o values yield a compromise between fewer localities and higher stability, we run AMF module extraction (step 3) $R_M = 10$ times for different numbers of modules M; we report reconstruction quality (main loss term as cosine similarity instead of distance), overestimation ratio, and silhouette width of AMF runs per M (Fig. 3b). ***End-to-end LAVA explanations***, or combined locality placement and AMF module extraction: we run AMF for $M = 9$ modules (criteria for M: Sec. 2.3) $R_M = 10$ times for each of the $R_L = 5$ locality placements; to assess similarity of the patterns underlying the modules irrespective of module organization (beyond silhouette width) across locality placements, we take a presence-weighted average of the module correlations per AMF run, average across runs per placement, and compare using cosine similarity (Fig. 3c); for an arbitrarily selected placement (Fig. 3c, green-highlight), we also compare between presence-weighted module averages of individual AMF runs (Fig. 3d). ***Sensitivity and ablation analysis*** for the remaining hyperparameters of locality placement (neighborhood size n) and module extraction (overestimation weight v, scale regularization strength γ) are in App. B.4.

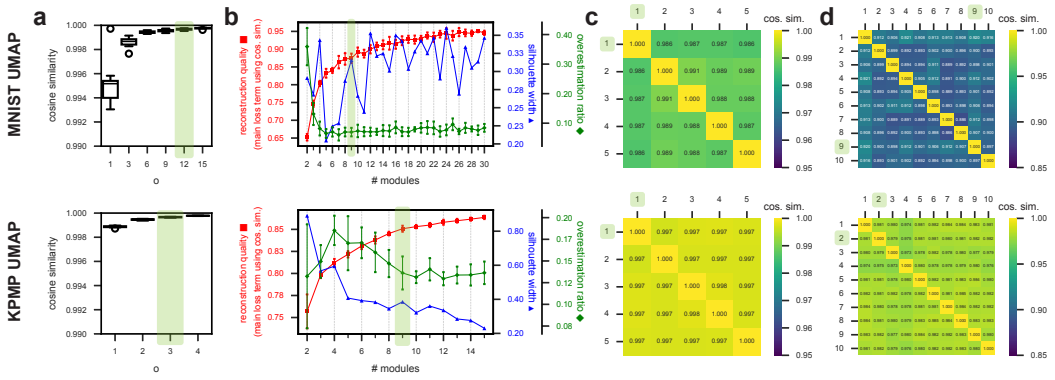


Figure 3: **LAVA stability.** Results for MNIST and KPMP UMAP embeddings. **a**, Cosine similarity of average correlations for locality placements with different overlap hyperparameter. **b**, Statistics of module extraction runs. **c-d**, Cosine similarity of presence-weighted averaged modules **c**, across locality placements; **d**, for one placement. Green bands denote selected hyperparameters and runs.

LAVA localities are stable. Feature-pair correlations are highly similar across locality placements for MNIST and KPMP (cosine similarity > 0.993), even with smaller numbers of localities influenced by overlap o, showing that localities are stable (Fig. 3a, App. B.5). **LAVA modules are stable.** As expected, larger number of modules M leads to better reconstruction (Fig. 3b, red). For MNIST, better reconstruction with small standard deviations (Fig. 3b, red), together with larger instability (smaller silhouette widths, Fig. 3b, blue), points to successful optimization with multiple valid solutions. For KPMP, a decrease in silhouette widths with increasing M (Fig. 3b, blue) indicates that smaller numbers of modules M result in higher stability. Just like for MNIST, similar trends in reconstruction and stability for KPMP suggest the existence of multiple valid solutions. Note that silhouette measures similarity of modules, which goes beyond capturing similar underlying patterns and could be too strict for complex data. We thus also assess pattern similarity (Fig. 3c-d, App. B.7). **LAVA delivers stable end-to-end explanations.** LAVA modules capture highly similar correlation patterns across the $R_L = 5$ locality placements (Fig. 3c), and across AMF runs for one placement

(Fig. 3d), indicating that LAVA explanations are stable end-to-end (steps 1 to 3). Together, high reconstruction quality and highly similar patterns (Figs. 3b-3d) show that AMF runs capture similar underlying patterns, representing equally valid solutions. **Sensitivity to hyperparameters n, ν, and γ.** Larger neighborhood n leads to better reconstruction for the same number of modules M, as expected since larger localities yield more general patterns (App. B.4). Overestimation weight ν leads to expected error ratios $\tau = \frac{1}{1+\nu}$. Larger ν results in sparser modules containing fewer correlation patterns from a larger number of localities, whereas low $\nu = 1$ produces averaged correlations as expected for a target overestimation of 50%. Scale regularization strength $\gamma = 0$ effectively ablates the term, leading to scale misalignment; at the other extreme, very large γ (e.g. $\gamma = 10000$) ablates the main loss term, resulting in much worse reconstruction and no control of overestimation. Generally, lower γ leads to larger overestimation, with partial scale control by the optimization.

3.2 Properties of LAVA Explanations

We assess if LAVA explanations exhibit the sought-after properties. **Setup.** We examine the LAVA modules obtained for UMAP embeddings of MNIST and KPMP, and for t-SNE embeddings of BOW (Sec. 3.1). **LAVA modules reveal globally reoccurring local patterns.** In all 3 cases, extracted modules represent globally reoccurring patterns not tied to neighboring localities (Fig. 4). For instance, the two MNIST modules with the highest sum of presences in the latent space capture patterns shared between digits 3, 5, and 8, and between digits 4, 7, and 9 (Fig. 4a). Similarly, for KPMP embeddings (Fig. 4b), some modules are present across most major cell types (module 3), while others pertain to more local regions and specific cell types (e.g. epithelial for modules 1, 2, and 4). Though BOW embeddings yield less structure, LAVA still identifies coherent modules relating to methodology from scientific text (correlations between words like 'number', 'point', and 'analysis'), and politics related to news text (words like 'government' and 'president'; App. B.9, Figs. S40-S43). Notably, LAVA explanations enable interpretation of structure that is not visually apparent without semantic document labels. Being spatially agnostic towards localities, LAVA identifies local patterns shared across (or varying within) distant regions and clusters in all 3 cases.

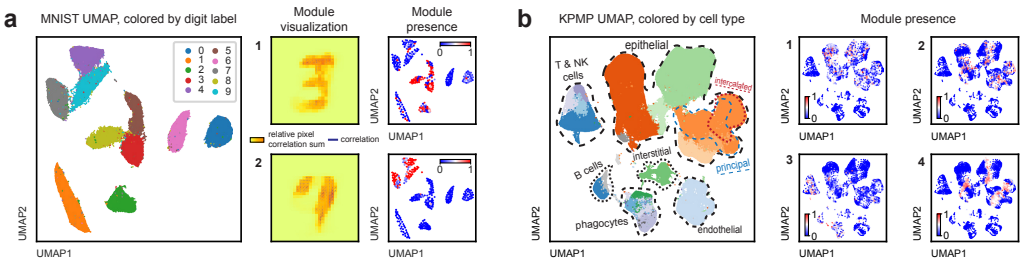


Figure 4: **LAVA explanations.** Modules & presences for UMAP embeddings: **a**, MNIST; **b**, KPMP.

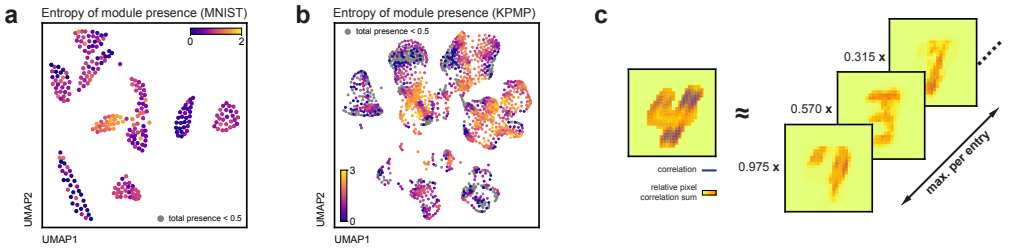


Figure 5: **LAVA module co-presences.** Co-presence as entropy of module presences per locality for UMAP embeddings of **a**, MNIST; **b**, KPMP. **c**, Example AMF decomposition of MNIST locality.

LAVA modules identify reoccurring patterns. Modules exhibit overestimation error ratios $< 15\%$ (Fig. 3b). Errors made up of predominantly underestimation indicate that locality-averaged patterns are avoided in favor of sparser, reoccurring patterns, as intended. Combined with the semantics of cosine similarity, presence of a module in two localities implies not only a shared pattern, but also that the pattern is directly present in both localities (Fig. 4, App. B.7). Comparing AMF to NMF

(App. B.10) reveals that, without overestimation control, NMF leads to 50% overestimation error ratio and denser NMF components than LAVA modules, representing average correlations rather than sparser reoccurring patterns across localities (Figs. S44- S47). With its reconstruction based on weighted sums, NMF does not adhere to the desired property that identified correlations are ensured to be present in individual localities, making them uninterpretable. **Multiple LAVA modules can coexist in a locality.** To examine co-presence of LAVA modules, we scale their presences to sum to 1 per locality and calculate Shannon entropy. Ignoring localities with non-scaled sums < 0.5 to prevent scale-inflated entropies, we get an average locality entropy of 0.68 (± 0.41) for MNIST and 1.41 (± 0.75) for KPMP (Figs. 5a-5b). For reference, 0 means no co-presence, and 3.17 corresponds to equal presence of all 9 modules. The obtained entropies indicate module co-presence in both datasets, especially KPMP (Figs. 5a-5c). Allowing co-presence, a locality can be related to different subsets of localities via presences of modules representing partially overlapping correlation patterns.

3.3 Comparing LAVA and Label-Free Explainability

We qualitatively compare LAVA to the label-free explainability framework (LFXAI; Crabbé & Schaar, 2022b). This is inherently challenging, since the methods are fundamentally different: LAVA is designed to explain unsupervised models, while LFXAI uses adaptations of XAI for supervised ML. We focus on LFXAI feature importance, as it relates input features to embeddings, and is conceptually closer to LAVA than example importance. Briefly, LFXAI explanations are per sample linear combinations of feature importances derived separately for each latent dimension as model output. By contrast, LAVA captures feature covariation patterns across embedding neighborhoods in the multidimensional latent space. **Setup.** We generate 4-dimensional MNIST embeddings with a denoising convolutional autoencoder, based on Crabbé & Schaar (2022b). We use LFXAI with gradSHAP (Lundberg & Lee, 2017) to get feature importances per sample. LAVA is run with numbers of neighbors and modules $n = 3000$ and $M = 8$, and overlap $o = 12$ (App. B.7.3). We directly inspect LFXAI/LAVA explanations of the autoencoder embeddings, with UMAP used only for visualization.

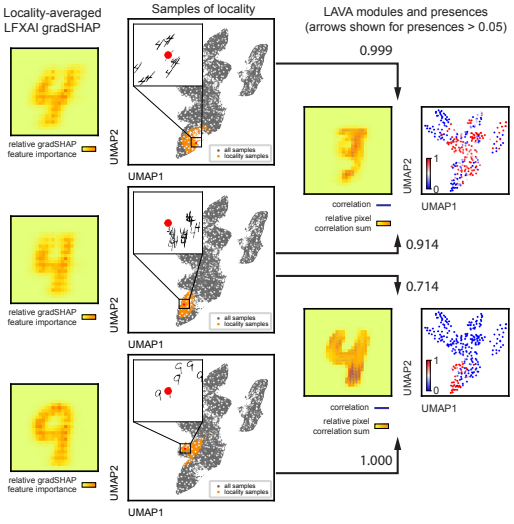


Figure 6: **LAVA & label-free XAI.** Example (left) LFXAI feature importances and (right) LAVA explanations – modules and presences – for localities of MNIST autoencoder embeddings.

Explaining samples vs. embedding structure. To enable comparison, we analyze explanations at LAVA locality level. We average LFXAI feature importances of the samples and compare to LAVA modules for the same locality (Fig. 6, App. B.7.4). The LFXAI locality-averaged importances resemble the original digit images (Fig. 6, LFXAI), consistent with the aim of revealing which features are relevant to place a sample in the latent space relative to a reference, empty image. However, it is nontrivial if and how LFXAI explanations can be leveraged to understand embedding structure, regardless of which importance method is used. In contrast, LAVA explains how input features relate to the embedding structure at a chosen level of granularity. By capturing reoccurring patterns across localities, LAVA can reveal relations between seemingly distant regions of the latent space that are not apparent otherwise (Fig. 6, LAVA). **LAVA enables discovery beyond attribution.** LAVA and LFXAI provide complementary explanations for latent embeddings, from different perspectives. Since one key goal of unsupervised learning is to identify patterns leading to new discoveries, the ability of LAVA to uncover hidden structure makes it ideally positioned to drive knowledge discovery beyond attribution (Figs. 4 and 6). Importantly, when relating features to embedding structure, LAVA detects feature covariation irrespective of feature usage by the model. This allows LAVA to include features otherwise ignored, for instance for exhibiting strong dependencies with others. As a result, LAVA delivers more complete explanations that address a common limitation of explainability methods (Molnar, 2022a), and circumvent the Rashomon effect (Breiman, 2001; Rudin et al., 2024).

3.4 EXPLAINING GENE EXPRESSION EMBEDDINGS

Finally, we assess if LAVA can reveal biologically relevant patterns in UMAP embeddings of KPMP data. **Setup.** We analyze LAVA modules for KPMP (Sec. 3.1), together with metadata on the human donors of the cells: we look at disease status, either healthy, acute kidney failure (ACF), or chronic kidney disease (CKD). To quantify the association between modules and a disease, we calculate Pearson's correlation and respective p-value between module presence and percentage of samples labeled with the disease (ignoring localities where a module is not present). We further investigate if genes in the modules are involved in biological pathways related to the disease (App. B.8).

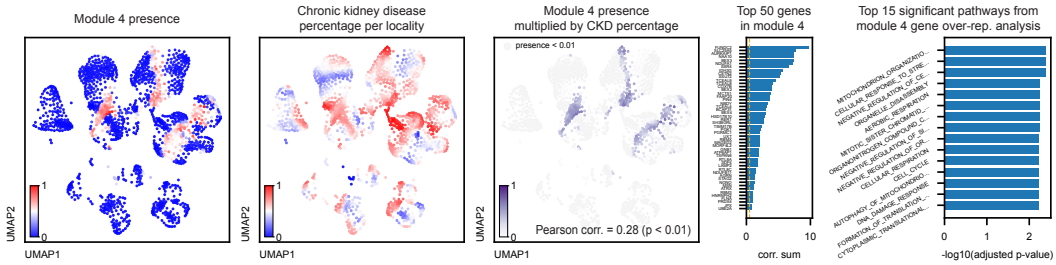


Figure 7: **LAVA module 4 for KPMP.** Module presence, chronic kidney disease (CKD) percentage, module-CKD overlap, gene analysis.

Module correlates with diseased samples. Cells from donors with distinct disease status are spread across the latent space, with no apparent clustering. However, presence of module 4 is specific to epithelial cells (Figs. 4b and 7) and correlates positively with percentage of diseased samples (0.23, $p < 0.01$), specifically CKD (Fig. 7, 0.28, $p < 0.01$), suggesting relevance for kidney disease. **Module genes and disease pathways.** Overrepresentation analysis of biological pathways (Fang et al., 2022) for module 4 identifies 21 significant pathways ($p < 0.01$). To highlight relevance, we single out "cellular response to stress" and "DNA damage response", previously linked to CKD (Patera et al., 2024; Schupp et al., 2016; Molitoris, 2019; Wang et al., 2023b). Our findings suggest that the local structure of KPMP UMAP embeddings exhibits cell type-specific patterns of gene correlation related to kidney disease. This local structure is not apparent in the clustering, but is notably exposed by LAVA modules. Identified genes and pathways can be used as candidates for future research.

4 CONCLUSION

We frame explainability of latent embeddings as enabling structural description of multivariate latent organization in terms of the underlying data. With this aim, we then propose LAVA, a model-agnostic method that generates prototype-like explanations, capturing local patterns of feature correlation reoccurring across the latent space. We show that LAVA is stable and complementary to existing attribution-based approaches. Notably, the explanations can uncover hidden structure not apparent otherwise, making it suitable to drive knowledge discovery. More broadly, we consider LAVA a proof-of-concept implementation of a strategy departing from conventional explainability for (un)supervised learning models: the focus shifts from answering why a given sample embedding is produced towards offering insight on patterns underlying the relation between spatial embedding structure and the original data. As such, the most influential parameter is locality size n, which steers the spatial granularity of explanations. We recommend trying multiple values to see what patterns emerge before settling, or setting n as the size of the smallest grouping apparent from the embedding structure (App. A.1.1). While LAVA can lead to multiple valid module structures, we show that recovered patterns are highly consistent. We leave this non-trivial overspecification for future work. Likewise, modalities such as text might benefit from some adaptation, e.g. changing locality representation (App. A.3). Beyond these, the principle of extracting higher-level commonality across finer-grained explanations could also be extended to supervised learning, complementarily to knowledge discovery based on model extrapolation. For instance, Shrikumar et al. (2020) identify transcription factors by extracting reoccurring motifs of DNA sequences from deep learning models. Developing new strategies to identify reoccurring patterns from models or embeddings of other modalities, and cross-modally, could yield new hypotheses in biology, medicine, and other domains.

AI USE STATEMENT

In this work, we did not use generative AI tools for any task. We take full responsibility for the final content of this work, including the text, the claims we make, and the code we provide.

ETHICS STATEMENT

In our experiments (Sec. 3), we use KPMP single-cell gene expression data, which contains information about human subjects. However, the data are publicly available and anonymized, and should therefore not constitute an ethical concern. Other than that, the developed method itself does not raise any ethical concerns other than general misuse applicable to any method.

REPRODUCIBILITY STATEMENT

We attempt to make our work as reproducible as possible through: (1) thoughtful experimental design; (2) a comprehensive appendix; and (3) publicly available code. Our experimental design takes into account stochasticity present in the LAVA method, meaning that any discrepancies that arise when reproducing results should not have an impact on the claims we make. Furthermore, the main text has references to the appendix to provide greater detail and resolve cases in which ambiguity might arise, both in methodology (App. A) and the experiments (App. B). Finally, all code will be made available on GitHub or another public repository, conditional on acceptance of the paper. It is currently attached as a .zip file to this submission.

REFERENCES

Ludmil B. Alexandrov, Serena Nik-Zainal, David C. Wedge, Peter J. Campbell, and Michael R. Stratton. Deciphering signatures of mutational processes operative in human cancer. *Cell Reports*, 3(1):246–259, 2013. ISSN 2211-1247. doi: 10.1016/j.celrep.2012.12.008. URL https://www.cell.com/cell-reports/abstract/S2211-1247(12)00433-0. Publisher: Elsevier.

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019. ISSN 1546-1696. doi: 10.1038/nbt.4314. URL https://www.nature.com/articles/nbt.4314. Publisher: Nature Publishing Group.

Claus Beisbart and Tim Räz. Philosophy of science at sea: Clarifying the interpretability of machine learning. *Philosophy Compass*, 17(6):e12830, 2022. ISSN 1747-9991. doi: 10.1111/phc3.12830. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/phc3.12830.

Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay. IXVC: An interactive pipeline for explaining visual clusters in dimensionality reduction visualizations with decision trees. *Array*, 11:100080, 2021. ISSN 2590-0056. doi: 10.1016/j.array.2021.100080. URL https://www.sciencedirect.com/science/article/pii/S259000562100028X.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi

Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL http://arxiv.org/abs/2108.07258.

Maria Brbić, Kaidi Cao, John W. Hickey, Yuqi Tan, Michael P. Snyder, Garry P. Nolan, and Jure Leskovec. Annotation of spatially resolved single-cell data with STELLAR. *Nature Methods*, 19(11):1411–1418, 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01651-8. URL https://www.nature.com/articles/s41592-022-01651-8. Publisher: Nature Publishing Group.

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1009213726. URL https://projecteuclid.org/journals/statistical-science/volume-16/issue-3/Statistical-Modeling--The-Two-Cultures-with-comments-and-a/10.1214/ss/1009213726.full.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018. doi: 10.1098/rsif.2017.0387. URL https://royalsocietypublishing.org/doi/full/10.1098/rsif.2017.0387. Publisher: Royal Society.

Richard Connor. A tale of four metrics. In Laurent Amsaleg, Michael E. Houle, and Erich Schubert (eds.), *Similarity Search and Applications*, pp. 210–217. Springer International Publishing, 2016. ISBN 978-3-319-46759-7. doi: 10.1007/978-3-319-46759-7_16.

Edouard Couplet, Pierre Lambert, Michel Verleysen, Dounia Mulders, John Lee, and Cyril De Bodt. Natively interpretable t-SNE. In *Proceedings of AIMLAI workshop*, volume 1, 2023. URL https://hdl.handle.net/2078.5/100421.

Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: a generalized framework for concept-based explanations. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 2590–2607. Curran Associates Inc., 2022a. ISBN 978-1-7138-7108-8.

Jonathan Crabbé and Mihaela van der Schaar. Label-free explainability for unsupervised models. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 4391–4420. PMLR, 2022b. URL https://proceedings.mlr.press/v162/crabbe22a.html. ISSN: 2640-3498.

Sebastian Damrich, Niklas Böhm, Fred A Hamprecht, and Dmitry Kobak. From t-SNE to UMAP with contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=B8a1FcY0vi.

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887):70–74, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-04086-x. URL https://www.nature.com/articles/s41586-021-04086-x. Publisher: Nature Publishing Group.

Zhuoqing Fang, Xinyuan Liu, and Gary Peltz. Gseapy: a comprehensive package for performing gene set enrichment analysis in python. *Bioinformatics*, 39(1):btac757, 11 2022. ISSN 1367-4811. doi: 10.1093/bioinformatics/btac757. URL https://doi.org/10.1093/bioinformatics/btac757.

Bas C. Van Fraassen. The pragmatic theory of explanation. In Joseph C. Pitt (ed.), *Theories of explanation*. Oxford University Press, 1988.

Henry Han, Wentian Li, Jiacun Wang, Guimin Qin, and Xianya Qin. Enhance explainability of manifold learning. *Neurocomputing*, 500:877–895, 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2022.05.119. URL https://www.sciencedirect.com/science/article/pii/S0925231222007044.

Doron Haviv, Ján Remšík, Mohamed Gatie, Catherine Snopkowski, Meril Takizawa, Nathan Pereira, John Bashkin, Stevan Jovanovich, Tal Nawy, Ronan Chaligne, Adrienne Boire, Anna-Katerina Hadjantonakis, and Dana Pe'er. The covariance environment defines cellular niches for spatial inference. *Nature Biotechnology*, pp. 1–12, 2024. ISSN 1546-1696. doi: 10.1038/s41587-024-02193-4. URL https://www.nature.com/articles/s41587-024-02193-4. Publisher: Nature Publishing Group.

Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Herbert B. Schiller, and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, 2023. ISSN 1471-0064. doi: 10.1038/s41576-023-00586-w. URL https://www.nature.com/articles/s41576-023-00586-w. Publisher: Nature Publishing Group.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 2004. doi: 10.5555/1005332.1044709. URL https://dl.acm.org/doi/10.5555/1005332.1044709.

Da Wei Huang, Brad T. Sherman, and Richard A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009. ISSN 0305-1048. doi: 10.1093/nar/gkn923. URL https://doi.org/10.1093/nar/gkn923.

Paul Humphreys. The philosophical novelty of computer simulation methods. *Synthese*, 169(3):615–626, 2009. doi: 10.1007/s11229-008-9435-2. Publisher: Springer.

D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993. ISSN 1573-2878. doi: 10.1007/BF00941892. URL https://doi.org/10.1007/BF00941892.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Publisher: Nature Publishing Group.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2668–2677. PMLR, 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL http://arxiv.org/abs/1412.6980.

Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13056-x. URL https://www.nature.com/articles/s41467-019-13056-x. Publisher: Nature Publishing Group.

Roger Koenker and Kevin F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, 2001. ISSN 0895-3309. doi: 10.1257/jep.15.4.143. URL https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143.

Blue B. Lake, Rajasree Menon, Seth Winfree, Qiwen Hu, Ricardo Melo Ferreira, Kian Kalhor, Daria Barwinska, Edgar A. Otto, Michael Ferkowicz, Dinh Diep, Nongluk Plongthongkum, Amanda Knoten, Sarah Urata, Laura H. Mariani, Abhijit S. Naik, Sean Eddy, Bo Zhang, Yan Wu, Diane Salamon, James C. Williams, Xin Wang, Karol S. Balderrama, Paul J. Hoover, Evan Murray, Jamie L. Marshall, Teia Noel, Anitha Vijayan, Austin Hartman, Fei Chen, Sushrut S. Waikar, Sylvia E. Rosas, Francis P. Wilson, Paul M. Palevsky, Krzysztof Kiryluk, John R. Sedor, Robert D. Toto, Chirag R. Parikh, Eric H. Kim, Rahul Satija, Anna Greka, Evan Z. Macosko, Peter V. Kharchenko, Joseph P. Gaut, Jeffrey B. Hodgin, Michael T. Eadon, Pierre C. Dagher, Tarek M. El-Achkar, Kun Zhang, Matthias Kretzler, and Sanjay Jain. An atlas of healthy and injured cell states and niches in the human kidney. *Nature*, 619(7970):585–594, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-05769-3. URL https://www.nature.com/articles/s41586-023-05769-3. Publisher: Nature Publishing Group.

Pierre Lambert, Rebecca Marion, Julien Albert, Emmanuel Jean, Sacha Corbugy, and Cyril De Bodt. Globally local and fast explanations of t-SNE-like nonlinear embeddings. 2022. URL https://hdl.handle.net/2078.5/229616.

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. ISSN 1558-2256. doi: 10.1109/5.726791. URL https://ieeexplore.ieee.org/document/726791. Conference Name: Proceedings of the IEEE.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. ISSN 1476-4687. doi: 10.1038/44565. URL https://www.nature.com/articles/44565. Publisher: Nature Publishing Group.

Zachary C. Lipton. The mythos of model interpretability, 2017. URL http://arxiv.org/abs/1606.03490.

Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. *Computer Graphics Forum*, 38(3):67–78, 2019. ISSN 1467-8659. doi: 10.1111/cgf.13672. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13672. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13672.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation, 2020. URL http://arxiv.org/abs/2010.14766.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. ISSN 1533-7928. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5.1, pp. 281–298. University of California Press, 1967. URL https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992.

Stavros Makrodimitris, Bram Pronk, Tamim Abdelaal, and Marcel Reinders. An in-depth comparison of linear and non-linear joint embedding methods for bulk and single-cell multi-omics. *Briefings in Bioinformatics*, 25(1):bbad416, 2024. ISSN 1477-4054. doi: 10.1093/bib/bbad416. URL https://doi.org/10.1093/bib/bbad416.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, 2020. URL http://arxiv.org/abs/1802.03426.

Marina Meilă and Hanyu Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11:393–417, 2024. ISSN 2326-8298, 2326-831X. doi: 10.1146/annurev-statistics-040522-115238. URL https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-040522-115238. Publisher: Annual Reviews.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 0004-3702. doi: 10.1016/j.artint.2018.07.007. URL https://www.sciencedirect.com/science/article/pii/S0004370218305988.

Bruce A. Molitoris. DNA damage response protects against progressive kidney disease. *The Journal of Clinical Investigation*, 129(11):4574–4575, 2019. ISSN 0021-9738. doi: 10.1172/JCI131171. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6819138/.

Christoph Molnar. *Interpretable Machine Learning*. 3rd edition, 2022a. URL https://christophm.github.io/interpretable-ml-book.

Christoph Molnar. Methods overview. In *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, chapter 4. 3rd edition, 2022b. ISBN 978-3-911578-03-5. URL https://christophm.github.io/interpretable-ml-book/overview.html.

Christoph Molnar. Prototypes and criticisms. In *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, chapter 26. 3rd edition, 2022c. ISBN 978-3-911578-03-5. URL https://christophm.github.io/interpretable-ml-book/proto.html.

Rajiv Movva, Peyton Greenside, Georgi K. Marinov, Surag Nair, Avanti Shrikumar, and Anshul Kundaje. Deciphering regulatory DNA sequences and noncoding genetic variants using neural network models of massively parallel reporter assays. *PLOS ONE*, 14(6):e0218073, 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0218073. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0218073. Publisher: Public Library of Science.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi: 10.1073/pnas.1900654116. URL https://www.pnas.org/doi/full/10.1073/pnas.1900654116. Publisher: Proceedings of the National Academy of Sciences.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*, chapter Manifold Learning, pp. 689–705. MIT Press, 2022. URL http://probml.github.io/book1.

Philipp Nazari, Sebastian Damrich, and Fred A. Hamprecht. Geometric autoencoders - what you see is what you decode. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 25834–25857. PMLR, 2023. URL https://proceedings.mlr.press/v202/nazari23a.html.

David Newman. Bag of Words. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5ZG6P.

Francesco Patera, Leonardo Gatticchi, Barbara Cellini, Davide Chiasserini, and Gianpaolo Reboldi. Kidney fibrosis and oxidative stress: From molecular pathways to new pharmacological opportunities. *Biomolecules*, 14(1):137, 2024. ISSN 2218-273X. doi: 10.3390/biom14010137. URL https://www.mdpi.com/2218-273X/14/1/137. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html. ISSN: 2640-3498.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL https://www.nature.com/articles/s42256-019-0048-x. Publisher: Nature Publishing Group.

Cynthia Rudin, Chudi Zhong, Lesia Semenova, Margo Seltzer, Ronald Parr, Jiachang Liu, Srikar Katta, Jon Donnelly, Harry Chen, and Zachery Boner. Amazing things come from having many good models, 2024. URL http://arxiv.org/abs/2407.04846.

Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, 2019. ISSN 1546-1696. doi: 10.1038/s41587-019-0071-9. URL https://www.nature.com/articles/s41587-019-0071-9. Publisher: Nature Publishing Group.

Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01434. URL https://doi.org/10.1162/neco_a_01434.

Myra L. Samuels. Simpson's paradox and related phenomena. *Journal of the American Statistical Association*, 88(421):81–88, 1993. ISSN 0162-1459. doi: 10.1080/01621459.1993.10594297. URL https://doi.org/10.1080/01621459.1993.10594297. Publisher: ASA Website _eprint: https://doi.org/10.1080/01621459.1993.10594297.

Nicole Schupp, Helga Stopper, and August Heidland. DNA damage in chronic kidney disease: Evaluation of clinical biomarkers. *Oxidative Medicine and Cellular Longevity*, 2016 (1):3592042, 2016. ISSN 1942-0900. doi: 10.1155/2016/3592042. URL https://onlinelibrary.wiley.com/doi/10.1155/2016/3592042. Publisher: John Wiley & Sons, Ltd.

Avanti Shrikumar, Katherine Tian, Žiga Avsec, Anna Shcherbina, Abhimanyu Banerjee, Mahfuza Sharmin, Surag Nair, and Anshul Kundaje. Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5.6.5, 2020. URL http://arxiv.org/abs/1811.00416.

Rory Stark, Marta Grzelak, and James Hadfield. RNA sequencing: the teenage years. *Nature Reviews Genetics*, 20(11):631–656, 2019. ISSN 1471-0064. doi: 10.1038/s41576-019-0150-2. URL https://www.nature.com/articles/s41576-019-0150-2. Publisher: Nature Publishing Group.

Ingo Steinwart and Andreas Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011. ISSN 1350-7265. doi: 10.3150/10-BEJ267. URL https://projecteuclid.org/journals/bernoulli/volume-17/issue-1/Estimating-conditional-quantiles-with-the-help-of-the-pinball-loss/10.3150/10-BEJ267.full. Publisher: Bernoulli Society for Mathematical Statistics and Probability.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023a. ISSN 1476-4687. doi: 10.1038/s41586-023-06221-2. URL https://www.nature.com/articles/s41586-023-06221-2. Publisher: Nature Publishing Group.

Peipei Wang, Jing Ouyang, Zhanjun Jia, Aihua Zhang, and Yunwen Yang. Roles of DNA damage in renal tubular epithelial cells injury. *Frontiers in Physiology*, 14:1162546, 2023b. ISSN 1664-042X. doi: 10.3389/fphys.2023.1162546. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10117683/.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization, 2021. URL http://arxiv.org/abs/2012.04456.

David S. Watson. On the philosophy of unsupervised learning. *Philosophy & Technology*, 36(2):28, 2023. ISSN 2210-5441. doi: 10.1007/s13347-023-00635-6. URL https://doi.org/10.1007/s13347-023-00635-6.

F. Alexander Wolf, Fiona K. Hamey, Mireya Plass, Jordi Solana, Joakim S. Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J. Theis. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biology*, 20(1):59, 2019. ISSN 1474-760X. doi: 10.1186/s13059-019-1663-x. URL https://doi.org/10.1186/s13059-019-1663-x.

Lucy Xia, Christy Lee, and Jingyi Jessica Li. Statistical method scDEED for detecting dubious 2d single-cell embeddings and optimizing t-SNE and UMAP hyperparameters. *Nature Communications*, 15(1):1753, 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-45891-y. URL https://www.nature.com/articles/s41467-024-45891-y. Publisher: Nature Publishing Group.

A METHODOLOGY

In this appendix, we further describe the methodology of our method.

A.1 LOCALITY DEFINITION

A.1.1 DEFINING THE SIZE OF LOCALITIES

The choice of hyperparameter n defines the granularity at which we are explaining the latent embeddings. This has an influence on the correlations LAVA will capture in step (2), and, subsequently, on the modules extracted in step (3).

A targeted approach in motivating the choice of n is using prior knowledge, e.g., known cluster labels or other metadata features. This can ground the scale of localities and make findings relevant for a specific downstream task (reoccurring intra- and intercluster correlations, correlations specific to specific metadata variable values, etc.). A more exploratory approach might be visualizing the latent embeddings and testing multiple values of n, to try to see what level of latent structure might contain meaningful patterns, which is especially relevant when there is little prior knowledge or metadata. Finally, n can also be chosen in a data-driven way, by looking at which size of latent neighborhood most resembles the neighborhoods in the original feature size (assuming that a relevant distance metric can be defined in the original feature space). In choosing such an n, we attempt to capture localities at a scale that most accurately represents organization in the original feature space, and the unsupervised model becomes a proxy for exploration of the original data.

In general, we can expect very small neighborhoods to produce noisy correlations that are artefacts rather than patterns, while very large values that cover a large portion of the dataset will fail to capture the structure of the latent organization and any meaningful reoccurring patterns within. Therefore, we advise against using both extremes as they are unlikely to be useful in the general case.

Choices for the datasets we use and additional experiments are in App. B.4.

A.1.2 OPTIMIZATION OF LOCALITY PLACEMENT

For locality placement, we minimize the difference between the relative in-degree centralities of samples when we look at the neighborhoods of all samples, and the neighborhoods of localities (see Sec. 2 and Equation 2). For example, if some sample is in the n-nearest neighborhoods of 1% of the samples in the embedding space, we want our localities to be placed so that the sample is a part of 1% of the localities, i.e., so that the sample retains its in-degree centrality. We do this by optimizing parameters α and β from Equation 2, utilizing the k-means (here referred to as ℓ-means) and DIRECT algorithms (Jones et al., 1993). If the number of localities is equal to the number of embedded samples ($l = E$), an optimal solution is one where the location of each embedding is the center of one locality (each sample is a center of a locality, its own n-sized neighborhood).

We use Manhattan instead of Euclidean distances in the loss function to focus less on outliers and approximate centrality more equally across the samples (equivalent to the difference between minimizing the mean absolute and mean squared errors). The rationale for using the weighing scheme in Equation 1 is making the computationally cheap ℓ-means algorithm to optimize our locality placement, i.e. minimize Equation 2. We want to weigh samples based on their centrality, since the ℓ-means algorithm would otherwise minimize Euclidean distances rather than our objective. Furthermore, the weights in Equation 1 factor in the inverse average distances to offset the fact that ℓ-means optimizes the sum of squared Euclidean distances between points, which means that lower density regions of the latent space might still have more probes (and localities), even when we weigh samples according to their centrality. Put together, we let the DIRECT algorithm optimize the weights of samples for the ℓ-means algorithm, such that the localities produced by the ℓ-means algorithm minimize our centrality-based loss function. Note that we use only the resulting centroids of the optimization as probes, while cluster assignments produced by the ℓ-means algorithm are left unused. The ℓ localities are then simply the n-sized neighborhoods around each of the ℓ probes.

Additionally, the DIRECT algorithm requires a range of values for the parameters in which the loss function is assumed to be convex. We fix this to $[-2D_L, 2D_L]$ for both α and β, where D_L is

the dimensionality of the latent space. We conjecture that, at least approximately (since ℓ-means is stochastic), the loss function is convex in that interval and contains a good solution. Loosely, as mentioned above, we want the weighing scheme to be able to overpower the overrepresentation of low-density regions that might occur if we would simply weigh samples based on their centrality. We argue about the range of values in terms of hyperball volume: the hyperball volume is proportional to the D_L-th power of its radius, which for a neighborhood of a sample we can approximate by the average distance to its neighbors, where D_L is the dimensionality of the latent space. To battle overrepresentation, we then want to weigh samples with larges "volumes" (samples in low density regions) inversely to their volume, so as to effectively force ℓ-means to give them less importance than they might otherwise have given the squared Euclidean distance optimization. In the two-dimensional case, the hyperball volume is simply the area of the circle, which is proportional to the square of the radius (the average distance of the n nearest neighbors). To allow weighing in such a fashion, we allow α to be at least 2, and we further multiply by 2 to allow for exaggeration of this principle. Since we cannot be sure how this weighing will interact with centrality weighing, we allow both parameters to also be negative, and give them the same range, arbitrarily. Through experiments, we show this yields consistent and relevant localities (e.g., Sec. 3 and Figs. 3, S21, and S34).

The pseudocode, omitting details relating to DIRECT, is shown in Alg. S1. We use the scipy[1] implementation of the DIRECT algorithm, and the scikit-learn[2] implementation of the k-means algorithm.

Algorithm S1 Locality Placement Algorithm.

require $\mathbf{X} \in \mathbb{R}^{E \times D_L}$, ℓ
initialize $nearest_neighbors[E, n]$, $in_neighborhood[E]$, $avg_n_distance[E]$
$nearest_neighbors \leftarrow$ calculate_nearest_neighbors$(\mathbf{X}, n)$
for $i = 1$ **to** E **do**
  $in_neighborhood[i] \leftarrow \frac{1}{E} \sum_{j=1}^{E} \mathbb{1}\left[\mathbf{X}_i \in nearest_neighbors[j]\right]$
  $avg_n_distance[i] \leftarrow \frac{1}{n} \sum_{j=1}^{n} ||\mathbf{X}_i - nearest_neighbors[i, j]||_2$
end for
initialize $localities[\ell, n]$, $probe_set[\ell]$, α_{opt}, β_{opt}
$\alpha_{opt}, \beta_{opt} \leftarrow$ locality_optimization$_{DIRECT}$[4]$(\mathbf{X}, \ell, in_neighborhood, avg_n_distance)$
$probe_set \leftarrow$ k_means$(\ell, \alpha_{opt}, \beta_{opt})$.centroids()
$localities \leftarrow$ calculate_nearest_neighbors$(probe_set, n)$

A.1.3 Defining the Number of Localities

To determine the number of localities (and respective probes), ℓ, we look at three factors: the number of embeddings E, the size of the neighborhood n, and an additional hyperparameter o relating to the overlap between localities. ℓ is then defined as:

$$\ell = [E * o/n], \tag{S1}$$

where $[\,]$ denotes rounding to the nearest integer. The hyperparameter o determines the number of localities given a dataset and neighborhood size n. If all samples are of a similar centrality (e.g., if data are uniformly spread-out across some space), $o = x$ would mean that each sample would be represented by approximately x localities. A larger o will produce more overlap between localities and enable a representation that more closely approximates the original in-degree centralities of samples, but will be more expensive computationally, without much added benefit. In Sec. 3 and Apps. B.5, we examine the stability of locality placement for UMAP embeddings of two different datasets in relation to hyperparameter o. Assuming stable (consistent) locality placement, we recommend using smaller values of o in the range of 3 to 30, depending on the size of the dataset, so that the resulting number of localities ℓ is not above several thousand, but still captures the various correlation patterns and how they change across the latent space.

[1] https://scipy.org/.
[2] https://scikit-learn.org/.
[6] More information in Sec. A.1.2.

A.2 Locality Representation

Correlations of each locality are calculated using the standard formulation of the Spearman correlation:

$$r_s(\mathbf{X}_i, \mathbf{X}_j) = \frac{cov\left[\mathrm{R}(\mathbf{X}_i), \mathrm{R}(\mathbf{X}_j)\right]}{\sigma_{\mathrm{R}(\mathbf{X}_i)}, \sigma_{\mathrm{R}(\mathbf{X}_j)}}, \tag{S2}$$

where X_i and X_j are two original input features and R is a ranking function. Covariance cov and standard deviations σ are estimated from the samples within a locality using standard estimators. Tied ranks are averaged.

Importantly, if one of the variables has the same value in 75% of the neighborhood, that correlation is set to 0 for that locality. This is done to prevent overestimation of correlations in cases where there is very little change. Notably, this has been the case for all of the datasets in our experiments, as the values of most pixels in MNIST are 0, and the gene expression at the single-cell level in KPMP (and more generally) is very sparse, and so are word counts. The value of 75% is set arbitrarily in order to allow for some leniency, while avoiding meaningless correlations based on low numbers of samples.

A.3 Alternatives to Locality Representation

In LAVA, localities are represented by the pair-wise correlations the features of their samples exhibit. However, the broader idea of extracting reoccurring patterns from the latent organization of data is amenable to different types of locality representation, which can be thought of as different inductive biases, uncovering different properties of embeddings. Changing representation will alter the basic idea of LAVA to explain latent structure through reoccurring patterns of input feature covariation, but will retain the broader notion of making latent organization more opaque by uncovering globally reoccurring local patterns in the embeddings. Here, we comment on several potential alternatives. Note that different locality representations will also likely require alternations to the subsequent module extraction step to produce interpretable results, which is outside of the scope of this work.

In Sec. 2, we mention why Spearman correlation is used instead of Pearson correlation. The simplest change to representation might be exploring different correlations, e.g., distance correlation, or other types of statistical relationships between variables.

Then, instead of statistical relationships, localities could also be represented by univariate statistical values, such as average values, quantile levels, or variances. Subsequently, modules could then identify regions of the latent space with subsets of variables that share some properties (are in a similar range, are highly variable, and so on).

In parametric models, localities could also be represented by feature importances extracted by the LFXAI framework or other methods. As many feature importances depend on a reference dataset or value, it might be interesting exploring using either the full dataset or only the samples of the locality as reference, as one characterizes a locality in relation to the entire embedding space, while the other characterizes it locally. Extracting reoccurring patterns of importance could then indicate reoccurring trends in how the model treats different parts of a dataset. Like in LAVA, distant parts of the latent space could also be treated similarly locally, indicating reoccurring variance and covariance in the data captured by the model.

In more complex modalities such as images and text, using original features such as words or pixels might be difficult or uninformative. Defining higher level features, by, for instances, measuring average activations of predefined concepts in each of the localities, could be useful. Modules might then reveal contexts where concepts co-occur, relating to work by Kim et al. (2018) and its numerous extensions, e.g., by Crabbé & Schaar (2022a).

A.4 Module Extraction

A.4.1 AMF loss function and hyperparameters

The AMF loss function (Equation 5) is discussed in Sec. 2.3. Here, we provide some additional detail and motivation.

Different metrics of (dis)similarity. We decide to use cosine distance for the main term of our loss function. However, a different metric (or divergence) for assessing multivariate (dis)similarity might be equally suitable, e.g., the Kullback-Leibler divergence. We did not find strong argumentation about whether the semantics of one of these might be more appropriate for our task than the other. Work relating to semantics and efficiency of these metrics suggests "triangular distance" as a more appropriate choice in some contexts (Connor, 2016), but is seldom used in contemporary literature.

Weighing the distance metric. We weigh cosine distance relative to the $L2$ norm of the correlations of a locality. Weighing in terms of scale makes sense for the scale-invariant cosine distance, as all localities would otherwise be treated equally, regardless of how much correlation they exhibit. We therefore assume that localities which exhibit no correlation have no valuable signal to extract commonality from, and thus should not have equal relevance for the loss function as localities whose samples have a clear and strong correlation patterns. Furthermore, we choose $L2$ norms rather than $L1$ norms, or some other alternative, so that optimization favors sparse, but high-value correlation pattern over low-value, spread-out patterns, which are more likely to be noise rather than signal.

Up-weighing overestimation and connection to quantile regression. The ν hyperparameter in Equation 5 determines how up-weighted overestimation errors will be in our loss. Weighing the errors is done similarly to how it is done in univariate quantile regression. The loss for univariate τ-th quantile regression, also known as the pinball loss (Steinwart & Christmann, 2011), is defined as:

$$\mathcal{L}_\tau(y, f(x)) = \begin{cases} \tau\,|y - f(x)| & \text{if } y \geq f(x) \\ (1-\tau)\,|y - f(x)| & \text{if } y < f(x) \end{cases}. \tag{S3}$$

Practically speaking, univariate quantile regression produces a weighing scheme between overestimation and underestimation errors: if $\tau > 0.5$, we are less concerned with overestimating y using $f(x)$, and vice versa. Using a $\tau = 0.5$ is equal to doing median regression, which corresponds to a regression that minimizes the mean absolute error, rather than the more standard mean squared error. For more intuition, we refer to Koenker & Hallock (2001) and Steinwart & Christmann (2011).

The idea for module extraction is to retain the semantics of similar, scale-invariant patterns we get with cosine distance or a similar metric, but allowing for reasoning about the reconstructed correlations in terms similar to quantile regression. Retaining the error weighing with a $\tau < 0.5$ is therefore done to prevent averaging across samples and extract only shared, reoccurring correlations (Sec. 2.3). Quantile-regression-like reconstructions will allow us to say that correlations present in module-based reconstructions are actually present in the localities with some certainty, or, in terms of errors, that the value of a correlation will be overestimated by at most τ (approximately). We can quantify whether this is actually the case by looking at the overestimation errors (Fig. 3b), which we calculate as the amount of overestimation errors divided by the total absolute error.

Our hyperparameter ν that weighs overestimation effectively corresponds to the ratio of weights for the overestimation and underestimation:

$$\nu = \frac{1-\tau}{\tau}. \tag{S4}$$

Using ν for weighing overestimation errors when using the mean absolute errors would be equivalent to doing τ-th quantile regression. We have set the hyperparameter $\nu = 9$ in all of our experiments. If we look at the pinball equivalent τ, this would correspond to 0.1-th quantile regression. We note that we cannot make direct parallels since we do not optimize the mean absolute error, but we can observe relative amounts of overestimation. These seem to approximately correspond to values we would expect if we would do quantile regression (Fig. 3b).

Extra: Presence fine-tuning. While we do not use it in our experiments, our implementation also includes what we call presence fine-tuning. After selecting modules, we can fix the module matrix $\mathbf{M}$ and optimize the presence matrix $\mathbf{P}$ with a pinball loss function (Equation S3), to guarantee the property that each locality (independent when $\mathbf{M}$ is fixed) has an overestimation ratio exactly equal to a chosen τ. Notice that the modules we extract are still contingent on the overestimation weighing used in the main loss term (because modules and presences are optimized jointly), which means that using weighing in the main loss term is not equivalent to first extracting the modules without any constraints and then fine-tuning at the end.

Scale regularization. The regularization term in Equation 5 is a simple mean absolute error term designed to retain scale of reconstructions. The idea behind scaling is simply to make reconstruc-

tions of the same scale as the actual locality correlations, as cosine distance is in itself scale invariant. While the cosine distance should remain relatively unaffected by this regularization, given a high enough hyperparameter γ, matching scale of the actual locality correlations and our reconstructions will allow us to observe the overestimation ratio of solutions (Sec. 3; Fig. 3b). Furthermore, due to the scale invariance of cosine distance, it is also imaginable that modules avoid overestimation simply by retaining low correlations values across the board: the scaling will push optimization away from such solutions. It is crucial that the scale regularization contains no overestimation weighing, meaning that the ratio of overestimation is purely a result of the up-weighing in the first term of the loss function, as this ensures the retainment of the semantics of reoccurring patterns, rather than focusing on absolute values of our solutions. We set $\gamma = 0.001$ as the default value for scale regularization to have a low impact on the loss function. We manually set this hyperparameter for the KPMP dataset to $\gamma = 10.0$ in the experiments (see Sec. B.4.2), as we noticed $\gamma = 0.001$ had no effect on optimization because of the low amounts of observed correlation. We do the same thing for the additional BOW dataset experiments in App. B.9.

Optimization hyperparameters. The matrices $\mathbf{P}$ and $\mathbf{M}$ are optimized to minimize the loss function (Equation 5) using gradient descent with minibatches of 64 samples (arbitrarily set as a standard value), the ADAM optimizer (Kingma & Ba, 2017) with default hyperparameters, and clamping. Clamping is used to enforce that any entries of matrices $\mathbf{P}$ and $\mathbf{M}$ are within the $[0, 1]$ interval after each step in training. Training is performed until convergence, which we define as no improvement in the loss function above 1% in 100 epochs. After convergence, we use the matrices from the iteration with the lowest loss as the final solutions as described in Sec. 3. We develop the code using the PyTorch library[7].

B EXPERIMENTS

In this appendix, we provide additional information about the main experiments and some additional results, such as hyperparameter sensitivity and runtime analyses.

B.1 SOFTWARE AND HARDWARE

All of our code uses Python 3.8 and its libraries[8]. We ran LAVA experiments on Linux compute nodes, comprising of a number of different CPUs and GPUs. We also tested all of our code locally, on an Apple M3 Pro, with 12 ARM CPU cores and 18GB of RAM. This machine also ran Jupyter notebooks containing result analyses, plot generation, and runtime experiments (App. B.11).

B.2 DATASETS, EMBEDDINGS, AND PREPROCESSING

B.2.1 MNIST

We downloaded the MNIST dataset using the Scikit-learn Python library[9] library. We performed no preprocessing. Prior to running LAVA, we generated: (a) a 2-dimensional UMAP embedding of the dataset using the UMAP[10] library, with default hyperparameters; (b) a 4-dimensional autoencoder embedding of the dataset using the denoising autoencoder from the Label-Free Explainability paper (Crabbé & Schaar, 2022b).

B.2.2 KPMP SINGLE-CELL KIDNEY ATLAS, V1.5

We downloaded the KPMP single-cell kidney atlas from the CELLxGENE website[11]. The dataset contains a UMAP embedding of the data preprocessed using the Seurat library[12], which we used

[7] https://pytorch.org/.

[8] Code included with supplementary material and will be made available online, conditional on acceptance.

[9] https://scikit-learn.org.

[10] https://umap-learn.readthedocs.io.

[11] https://cellxgene.cziscience.com. Specific version: https://datasets.cellxgene.cziscience.com/f5b6d620-76df-45c5-9524-e5631be0e44a.h5ad.

[12] https://satijalab.org/seurat/.

for our experiments. More information can be found at the KPMP data portal website[13]. The UMAP of the data can also be explored on the CELLxGENE website[14]. For module extraction, we remove mitochondiral, ribosomal, and mitochondrial ribosome genes. Then, we select the top 1000 most variable genes, based on the available gene expression data (data after normalization and log transformation).

B.3 Visualizations of correlations for the MNIST dataset

The MNIST dataset consists of images that are 28 pixels high and wide, totaling 784 pixels. To show the importance of each feature in a correlation matrix, we create heatmaps of the same dimensionality (28×28), where the value of a pixel correspond to the sum of the correlations that pixel participates in. Coloring is relative, meaning that it conveys relative importance of pixels for one locality or module (and its corresponding correlation matrix) only, and not across different visualizations. Furthermore, we add lines to connect pixels that are correlated to one another. This helps visualize the correlations which cause pixels to be important in the first place. To avoid clutter, we exponentiate correlations to the power of 3, and then remove correlations that are lower than 0.1 after exponentiation (arbitrarily, for visual clarity). The widths of lines that represent correlations are proportional to their exponents to differentiate between stronger and weaker correlations. Some additional examples of visualizations can be seen in Fig. S1.

Intuitively, we can think about these visualizations as a way to visualize pixel movement in a group of samples. In the case of local groups of similar images of digits, like we had in our experiments with MNIST, we can assume correlations will most likely reflect changes in pixel intensity due to minor translations, rotations, stretchings, or shrinkings of images. First, for all of these transformations, we can expect that nearby pixels will be correlated, which we can see in all visualizations (Fig. S1). For the latter three transformations, we can also expect further apart pixels to show patterns of correlation, which we see more clearly in correlations relating to the digits 0 and 1, as well as some other localities.

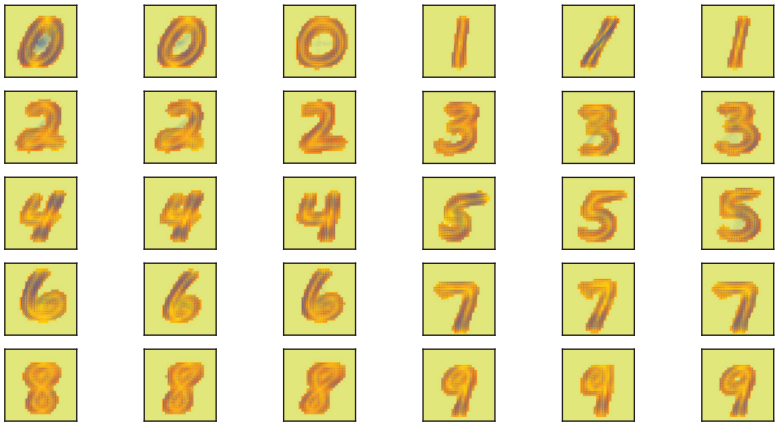


Figure S1: Example visualizations of correlations at various localities from our experiments with the MNIST dataset.

[13] https://atlas.kpmp.org/.

[14] https://cellxgene.cziscience.com/e/dea717d4-7bc0-4e46-950f-fd7e1cc8df7d.cxg/.

B.4 LAVA HYPERPARAMETERS

B.4.1 SIZE OF LOCALITY, n

Here we discuss and experiment with setting the hyperparameter n, which defines the granularity of the latent localities we explain.

Data-driven approach of setting n (MNIST)

A data-driven way of choosing the value n can be looking at the Jaccard similarity of neighborhoods of varying sizes in the original space and the latent space. Even though methods such as UMAP have hyperparameters that represent the size of neighborhoods that will be optimized for, different neighborhood sizes might end up being optimized (depending on the choice of other hyperparameters and the dataset), and these neighborhoods will often differ significantly from the original space. For the UMAP of the MNIST dataset we us in our experiments (Sec. 3), we find that with the default hyperparameter of 15 nearest neighbors, the resulting embedding neighborhoods is most representative of the neighborhoods in the original feature space when $n \approx 3000$ (Fig. S2), which is the value we use for the locality size hyperparameter in our experiments with MNIST. For simplicity, we also retain $n = 3000$ for the LAVA experiments with the autoencoder-generated embeddings.

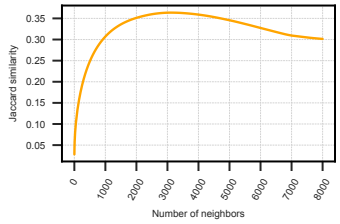


Figure S2: Average Jaccard similarity between original and latent space neighborhoods, calculated for a UMAP of the MNIST dataset with default parameters. Despite the default neighborhood size parameter being 15, we found the highest original neighbor retention to be around 3000 neighbors.

Sensitivity analysis (MNIST)

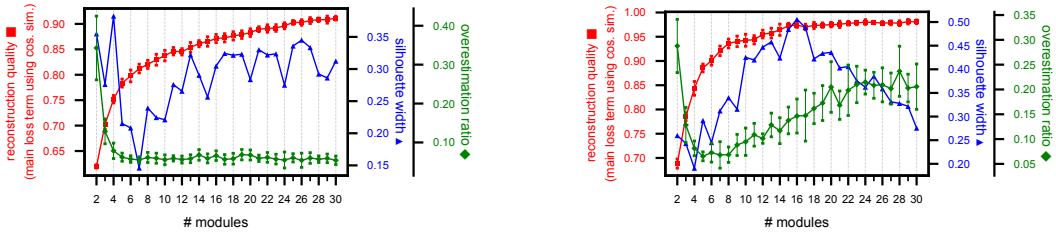
We also perform a sensitivity analysis of the hyperparameter for the UMAP MNIST embeddings. Other than $n = 3000$, we also look at localities of size $n = 500$ (with $o = 6$) and $n = 6000$ (with $o = 24$), not changing other hyperparameters.

Like in Fig. 2a, we report the results of module extraction across different values for M in Fig. S3. In general, we see that for the same number of modules, reconstruction quality is lower when the sizes of neighborhoods are smaller, and vice versa. We hypothesize that this can be expected in general, as larger localities will cover larger portions of a dataset. For instance, in the case of of MNIST, there are 70000 samples and we can imagine $10 - 20$ LAVA localities of size $n = 6000$ to cover most embeddings of the entire dataset. Then, extracting modules that correspond to the correlations of each of these localities will yield a very high reconstruction quality, and increasing the number of localities will just produce a larger number of localities with similar due to a large number of overlapping samples. In other words, there is a smaller number of patterns to extract from. With smaller localities, the same number of modules will be forced to extract the reoccurring patterns (if there are any) from a larger number of potentially different, more specific patterns, yielding a lower reconstruction quality.

Then, as expected, for $n = 6000$ we observe the reconstruction quality quickly reaches values close to 1 when increasing M. Interestingly, the loss function then starts to produce a tradeoff between cosine similarity and the overestimation error, and the reconstruction quality starts increasing at the expense of the overestimation error. The fact that this does not happen for lower values of reconstruction quality (higher values of the loss) indicates that the tradeoff does not exist for lower reconstruction qualities, i.e., the increase in reconstruction quality is not at odds with the overestimation regularization, which only restricts the space of possible solutions in those cases. We examine the regularization hyperparameters further in App. B.4.2.

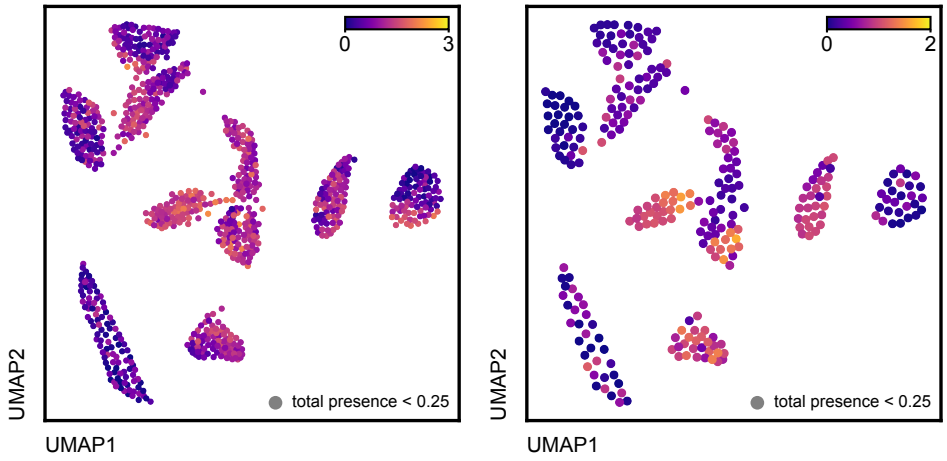
Furthermore, we also look at the module presence entropy like in Sec. 3 (Fig. S4), to see how much the extracted modules overlap for differently sized localities. The average entropy of the presences is 0.935 (± 0.497) for $n = 500$, 0.684 (± 0.415) for $n = 3000$ (see Fig. 5), and 0.551 (± 0.417) for $n = 6000$. This confirms our previous hypothesis that, for the same number of modules, smaller sized localities will produce more specific correlation patterns and modules will be reoccurring patterns found across a larger number of localities, subsequently forced to overlap (cooperate) to reconstruct these localities more accurately.

Finally, we also visualize the modules and their presences when $M = 9$, to compare them qualitatively (Figs. S5 and S6). For smaller localities, most patterns are more sparse and more spread out, with relatively continuous presence values, while the bigger localities have fairly binary presences, reflecting the clustering structure of the MNIST embeddings, which makes localities in these clusters almost identical to one another. These results are consistent with what we already observe quantitatively.



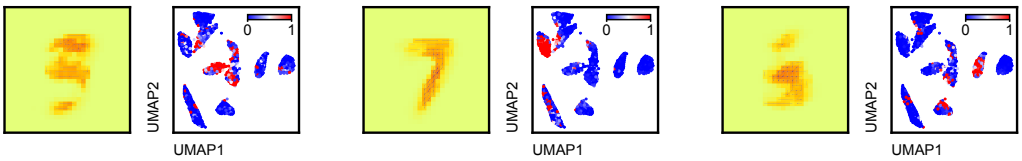
(a) $n = 500$ (b) $n = 6000$

Figure S3: Module extraction runs for the MNIST UMAP with different values of n.

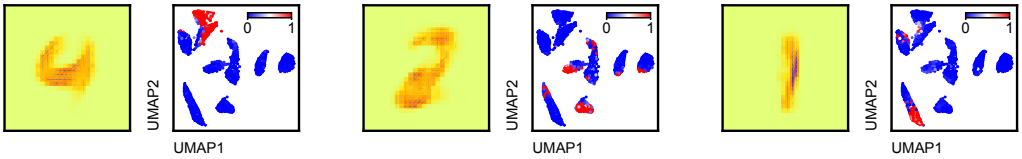


(a) $n = 500$ (b) $n = 6000$

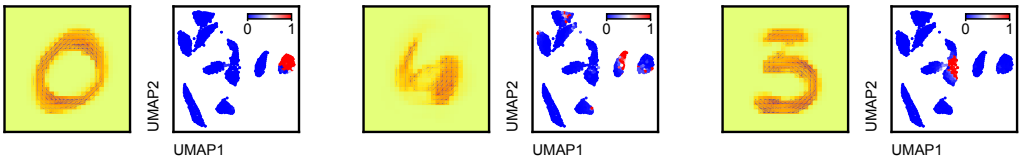
Figure S4: Module presence entropy across localities for the MNIST UMAP with different values of n.



(a) Module 1. (b) Module 2. (c) Module 3.

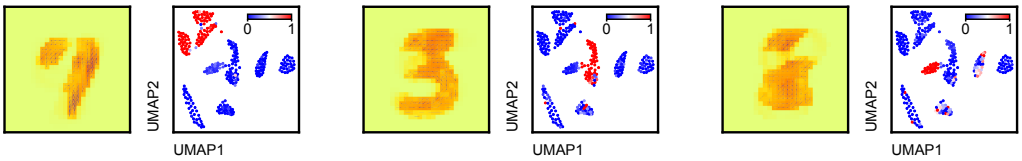


(d) Module 4. (e) Module 5. (f) Module 6.

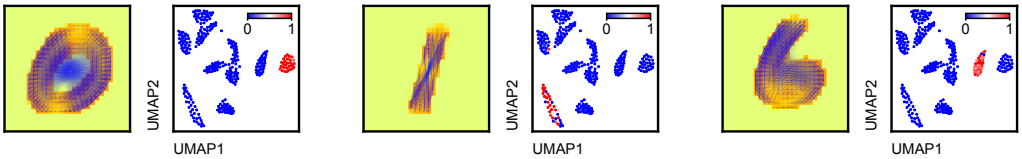


(g) Module 7. (h) Module 8. (i) Module 9.

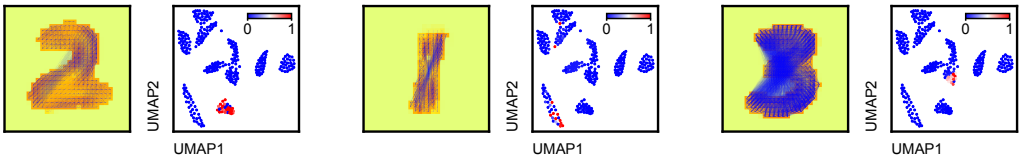
Figure S5: Visualizations and presences of the 9 extracted modules for MNIST when $n = 500$.



(a) Module 1. (b) Module 2. (c) Module 3.



(d) Module 4. (e) Module 5. (f) Module 6.



(g) Module 7. (h) Module 8. (i) Module 9.

Figure S6: Visualizations and presences of the 9 extracted modules for MNIST when $n = 6000$.

Meta-data approach for setting n (KPMP and BOW)

For the KPMP dataset, we are interested in observing patterns of gene correlation on a sub-cell type level. Looking at cell types in the data, we observe the median number of cells for each cell type is 3592, while the smallest cell type has 148 cells (Table S1). Accordingly, we chose neighborhoods of size $k = 500$, as this seemed like a large enough number to capture correlations between gene expressions that are unlikely to be noise and reflect latent structure, while small enough to explore differences within most of the cell type clusters.

Table S1: Quartile statistics of the numbers of samples per cell type.

	min.	25%	50%	75%	max.
# cells	148	1665	3592	9105	54025

For the BOW dataset, we use an analogous approach. The BOW dataset is comprised of 5 corpora, containing 1500 NeuRIPS articles, 3530 KOS blog entries, and a random subsample of 10000 documents from the three other corpora. A neighborhood size of $k = 500$ will thus produce cluster-specific localities, as well localities that can capture potential variation within and between those clusters. The smaller size of the localities is further motivated by the fractured structure of the t-SNE embeddings (see App. B.9, specifically Fig. S39a).

B.4.2 Module extraction hyperparameters

Here, we perform a sensitivity analysis of the overestimation error regularization weight ν and the scale regularization strength γ hyperparameters.

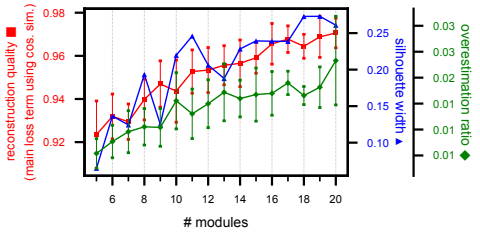
Overestimation regularization weight ν sensitivity analysis (MNIST)

The interpretation of the ν parameter relates to univariate quantile regression, where $\tau = \frac{1}{1+\nu}$, but substituting the usual mean absolute error for the cosine similarity (as described in Sec. 2.3 and App. A.4). To evaluate these effects, we run module extraction on the MNIST UMAP like in Sec. 3, varying the value of ν and keeping all other hyperparameters constant. In addition to the default $\nu = 9$, we test $\nu = 99, 19, 3, 1$, values which would correspond to $\tau = 0.01, 0.05, 0.25, 0.50$ univariate quantile regressions (Fig. S7).

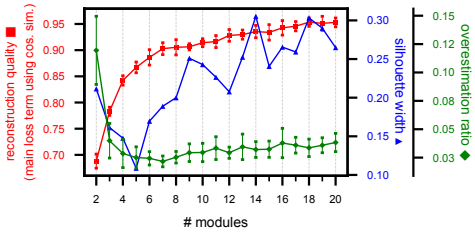
We observe overestimation error ratios approximately correspond to the τ values, being around or a bit above 0.01 when $\nu = 99$, 0.05 when $\nu = 19$, and a bit above the expected 0.25 for $\nu = 3$. When $\nu = 1$, no weighing takes place, meaning overestimation or underestimation are equivalent, and solutions will depend on the dataset: we observe overestimation error ratios of about 60%.

Then, looking at the stability of modules via silhouette width, we can see that using larger ν values will results in less stable module extraction. When put into relation with the relatively low reconstruction quality standard deviations, this seems to indicate that higher overestimation weighing produces a loss landscape with a larger number of equally valid solutions. We also observe much higher values of reconstruction quality when increasing ν. This is because reconstruction quality is measured as 1 minus the loss term, meaning that ν will have an impact the measured values, and the graphs are not comparable to one another. We measure the $L2$-weighed cosine similarity without overestimation weighing (Fig. S8) to compare between them more directly, and then see the opposite trend, as we would expect when thinking about overestimation weighing as a constraint or regularization.

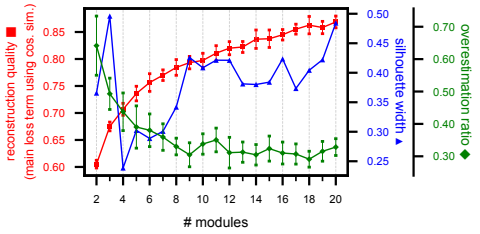
Finally, when qualitatively observing the differences in extracted modules of the extreme $\nu = 99$ and $\nu = 1$ (Figs. S9 and S10), we also see the expected behavior: when overestimation is weighted highly, the resulting modules are very sparse, more carefully selecting the correlations that are highly present in a larger number of localities, while $\nu = 1$ produces averaged correlations across, which we can also see in the more continuous presences. Importantly, without overestimation weighing, it is unclear



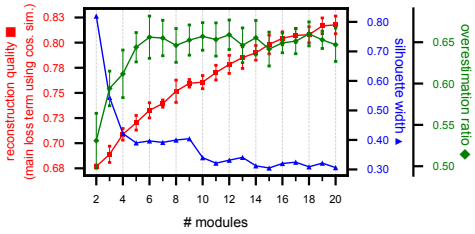
(a) $\nu = 99$



(b) $\nu = 19$

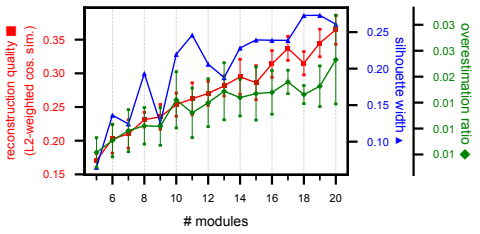


(c) $\nu = 3$

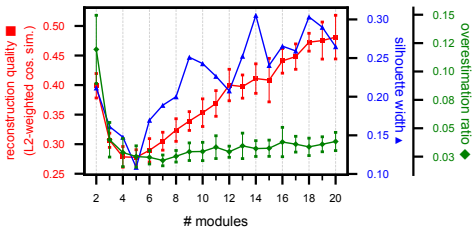


(d) $\nu = 1$

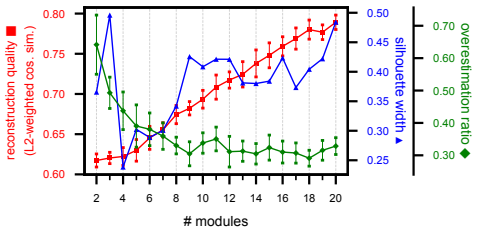
Figure S7: Module extraction sensitivity analysis for hyperparameter ν.



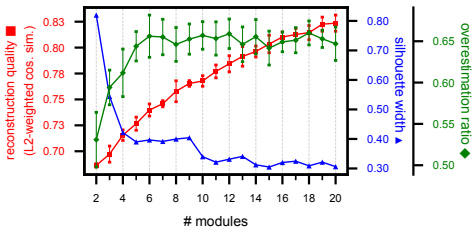
(a) $\nu = 99$



(b) $\nu = 19$

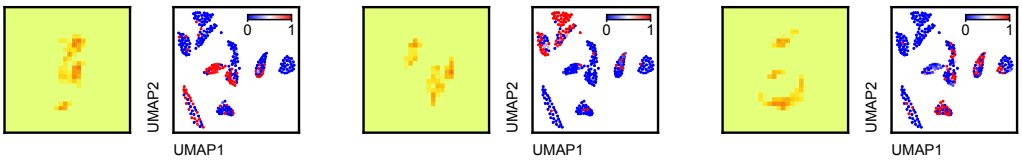


(c) $\nu = 3$

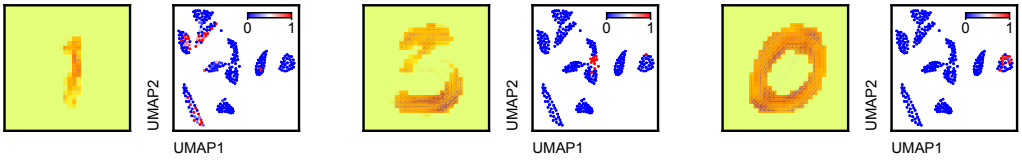


(d) $\nu = 1$

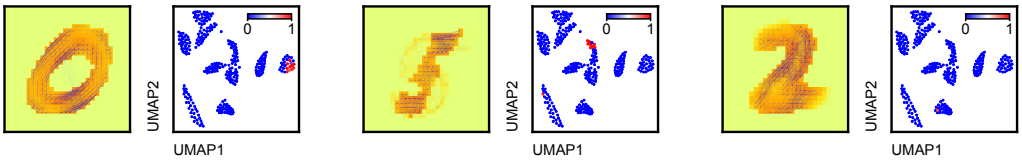
Figure S8: Module extraction sensitivity analysis for hyperparameter ν, where reconstruction quality is measured as the $L2$-weighted cosine similarity without overestimation weighing.



(a) Module 1. (b) Module 2. (c) Module 3.

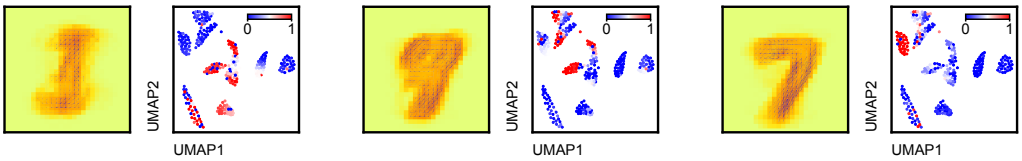


(d) Module 4. (e) Module 5. (f) Module 6.

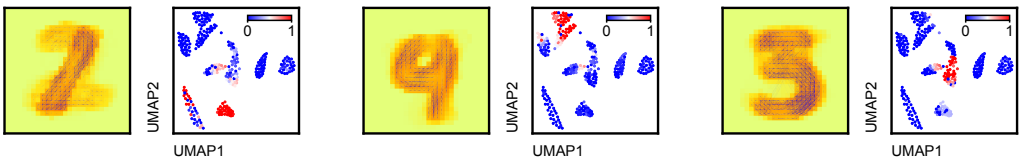


(g) Module 7. (h) Module 8. (i) Module 9.

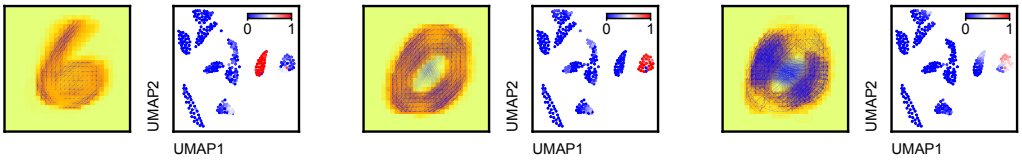
Figure S9: Visualizations and presences of 9 extracted modules for MNIST UMAP with $\nu = 99$.



(a) Module 1. (b) Module 2. (c) Module 3.



(d) Module 4. (e) Module 5. (f) Module 6.



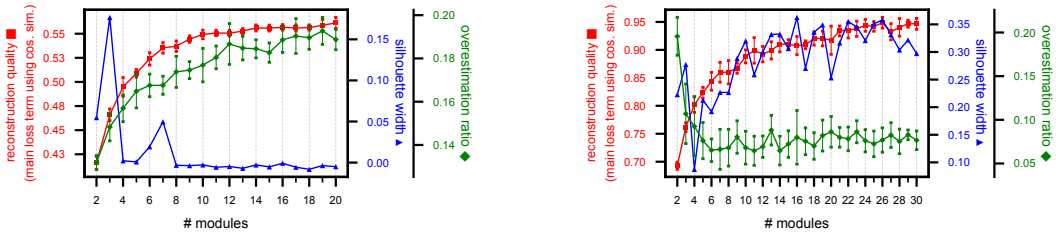
(g) Module 7. (h) Module 8. (i) Module 9.

Figure S10: Visualizations and presences of 9 extracted modules for MNIST UMAP with $\nu = 1$.

Scale regularization weight γ sensitivity analysis (MNIST & KPMP)

Here, we perform sensitivity analysis for the γ scale regularization hyperparameter on the UMAP embeddings of the MNIST dataset. Scale regularization should not have a large impact on the loss function, but is required to produce modules that are interpretable, and for which we can observe the overestimation error accurately (more in Sec. 2 and App. A.4).

Other than the default $\gamma = 0.001$ used in Sec. 3 (Fig. 3), we run AMF with $\gamma = 0$ and $\gamma = 10000$, where $\gamma = 0$ corresponds to the ablation of the scale regularization term, and $\gamma = 10000$ effectively corresponds to ablation of the main loss term (Fig. S11). For $\gamma = 10000.0$, we see that the main loss term is mostly ignored, with no control of overestimation and much lower reconstruction quality. For $\gamma = 0.0$, when compared to the $\gamma = 0.001$ in Sec. 3, we see a larger overestimation, indicating that the scale is partially kept in check by the optimization (clamping values to the $[0, 1]$ interval), but is inaccurate in relation to the default value.



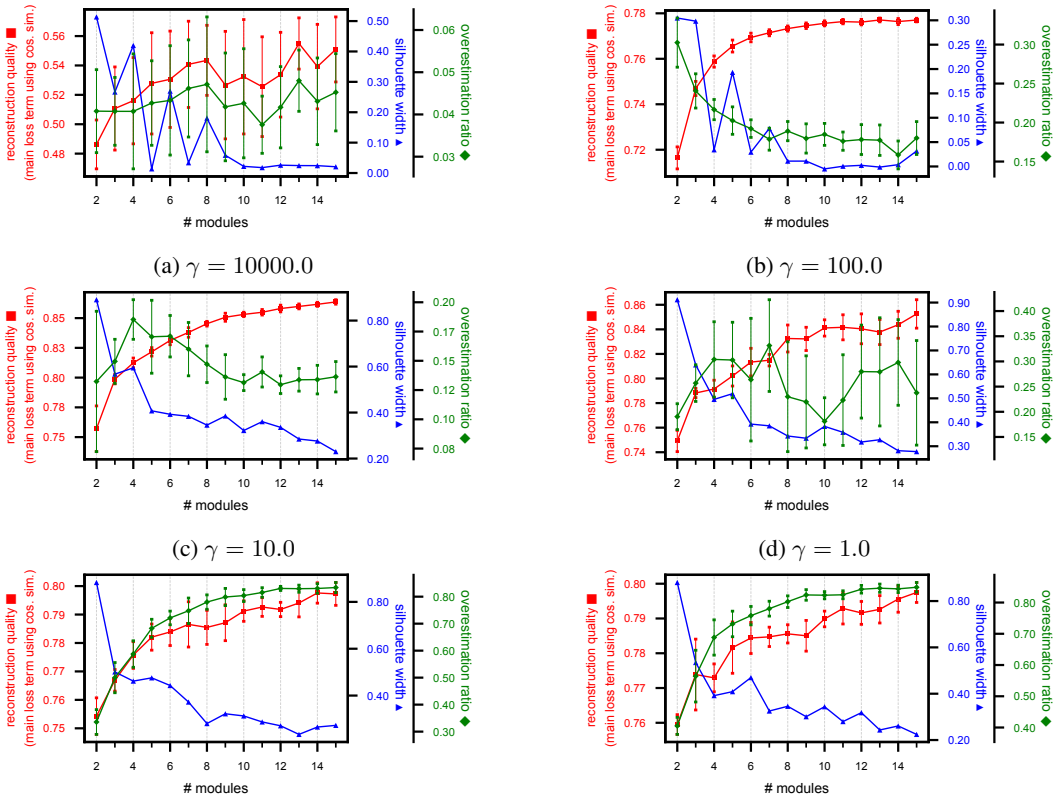
(a) $\gamma = 10000.0$ (b) $\gamma = 0.0$

Figure S11: Module extraction runs of the MNIST UMAP embeddings with different values of regularization hyperparameter γ.

Likewise, for the $KPMP$ dataset, we fix other hyperparameters like in Sec. 3 and perform a sensitivity analysis of γ. We see similar trends. In KPMP, however, we notice that using the default value of $\gamma = 0.001$ results in high overestimation error ratios. This points to the fact that regularization is probably not affecting the loss function and that the scale of the modules is not accurate, which we hypothesize to be the case due to the sparsity of gene expression, and subsequent low correlations in many localities that result in too small MAEs (Eq. 5). Running module extraction several times, using higher values of γ, confirms this hypothesis (Fig. S12). Increasing the strength of regularization seems to have no or negative effect on the quality or stability of solutions (the main loss term and silhouette width, respectively) until $\gamma = 100.0$, when MAE starts dominating the loss function, resulting in lower reconstruction quality and higher overestimation errors. With $\gamma = 10000.0$, the cosine distance loss is again effectively ablated, resulting in lower reconstruction quality and no control of overestimation. For our experiments, we end up using $\gamma = 10.0$, resulting in relatively low and stable error overestimation ratios (Fig. S12a).

B.5 Locality Placement

Here, we show some additional results for experiments relating to locality placement. The experiments were done using the UMAP embeddings of the MNIST and KPMP dataset. In the main paper (Fig. 3a), we show how similar the average locality correlations are across 5 different locality placement runs while varying the hyperparameter o. For more qualitative comparison, here we also show at the specific probe placements for the chosen hyperparameters $o = 12$ and $o = 3$ (for MNIST and KPMP, respectively) in Fig. S13 and Fig. S14, respectively. While we do see minor differences, the number of probes across the different areas and their locations remain relatively similar. Furthermore, our results (Fig. 3a, specifically the hyperparameters highlighted in green) indicate that the correlation patterns they capture are extremely similar. To further substantiate this claim, we also show cosine similarities of the average locality correlations across the 5 different runs in Fig. S15.



(a) $\gamma = 10000.0$

(b) $\gamma = 100.0$

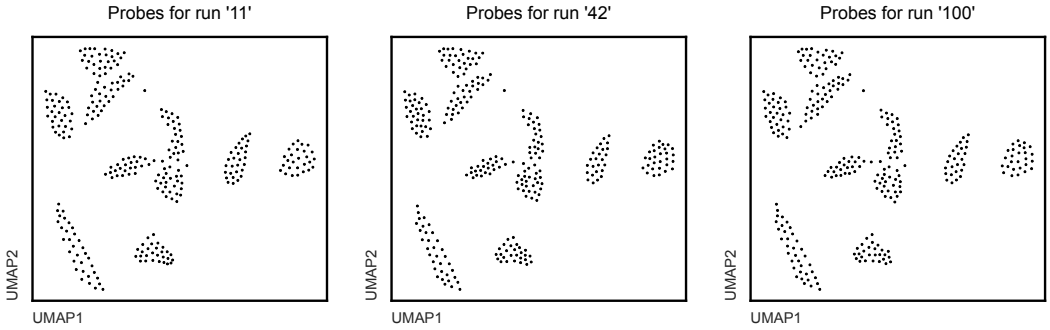
(c) $\gamma = 10.0$

(d) $\gamma = 1.0$

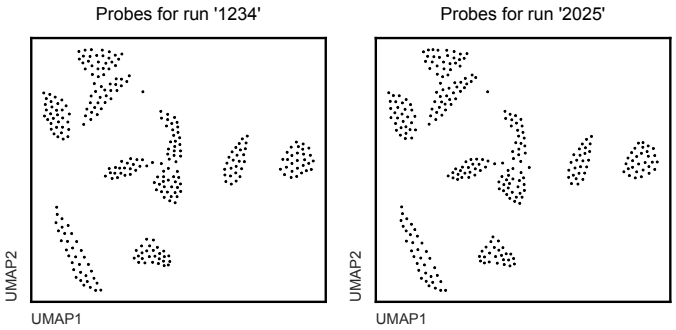
(e) $\gamma = 0.01$

(f) $\gamma = 0.001$

Figure S12: Module extraction runs of the KPMP UMAP embeddings with different regularization hyperparameters γ.

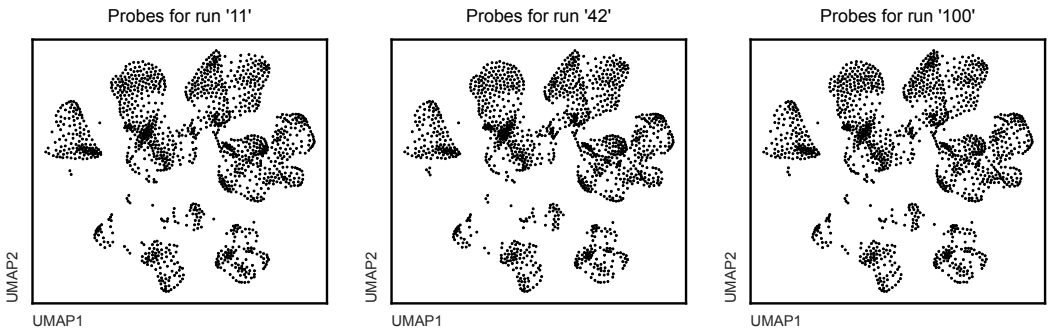


(a) Locality (probe) placement 1. (b) Locality (probe) placement 2. (c) Locality (probe) placement 3.

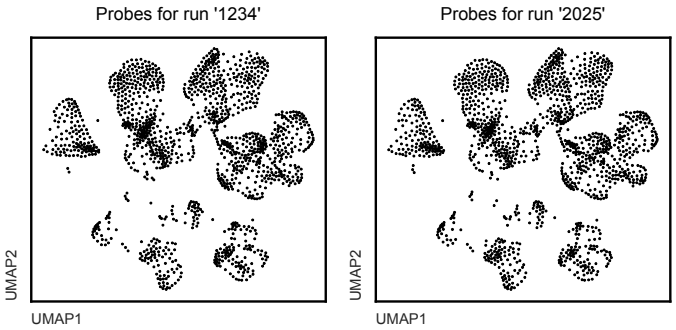


(d) Locality (probe) placement 4. (e) Locality (probe) placement 5.

Figure S13: The 5 different probe placements for the UMAP MNIST embeddings, $o = 12$.

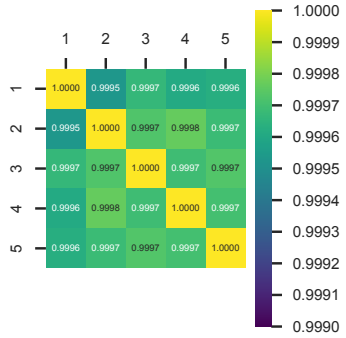


(a) Locality (probe) placement 1. (b) Locality (probe) placement 2. (c) Locality (probe) placement 3.

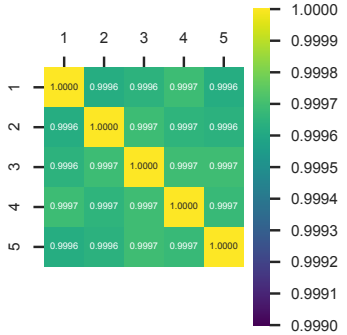


(d) Locality (probe) placement 4. (e) Locality (probe) placement 5.

Figure S14: The 5 different probe placements for the UMAP KPMP embeddings, $o = 3$.



(a) Cosine similarity between the average locality correlations for the MNIST UMAP embeddings, $o = 12$.



(b) Cosine similarity between the average locality correlations for the KPMP UMAP embeddings, $o = 12$.

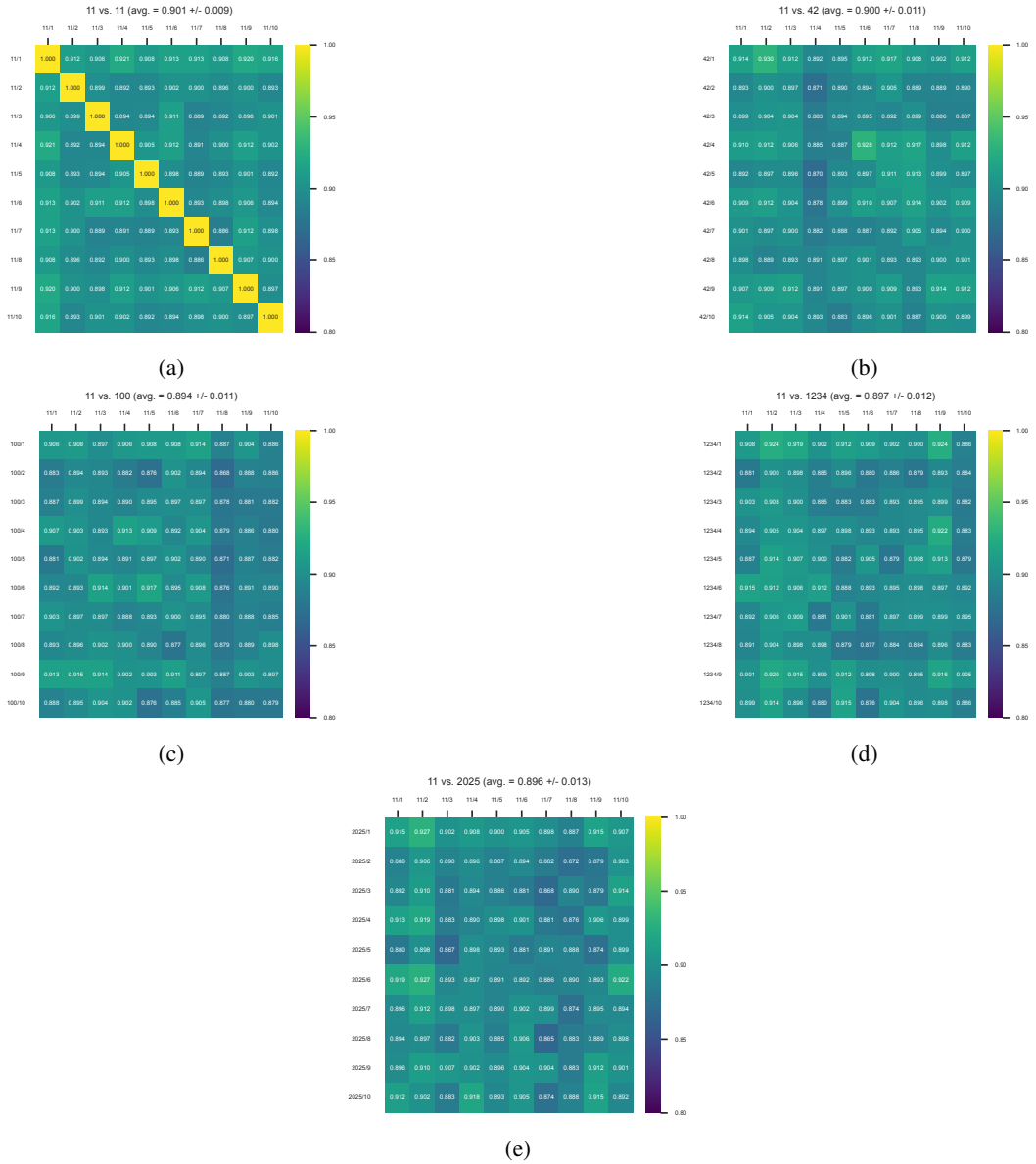
Figure S15: Cosine similarities across the 5 different locality placement runs for the MNIST and KPMP UMAP embeddings. Locality placements are named by the integers used as seeds for the different runs.

B.6 End-to-End Module Extraction Stability

Here, we show some additional results relating to module extraction stability, to reinforce the case that: (a) LAVA can produce consistent and stable results; (b) the instability of module extraction in our experiments is caused by the multiplicity of possible factorizations (overspecification), rather than algorithmic instability of our methodology. These results are based on LAVA runs on the UMAP embeddings of the MNIST and KPMP datasets from Sec. 3.

To expand on results from Fig. 3c, instead of averaging the correlation patterns the modules extract across different locality placements like we did there, here we show the comparisons of the presence-weighted module averages between all of the module runs, separately, from the chosen locality placement (denoted by its random seed 11) and the other 4 locality placement runs (Fig. S16 for MNIST, and Fig. S17 for KPMP). The high cosine similarities indicate that the correlation patterns that module extraction explains are very similar across the different locality placements, even between individual module extraction runs, further highlighting the stability of our solutions. We also note that the cosine similarities between the presence-weighted module averages within one locality placement run are not significantly different than across the locality placement runs (e.g., compare figures S16a and S16c).

Furthermore, we also look at the end-to-end stability of the module extraction when we change the overlap parameter o as we did in Fig. 3a. For each of the parameters o, we take averages of the presence-weighted module average across the 10 runs for a single locality placement, and then compare between the different values of o (Fig. S18). These results further expand results demonstrate that the similarities of the captured correlations are not high only within different locality placements for some hyperparameter o, but also across different hyperparameters, and that these correlations are preserved and explained by our module extraction.



(a) (b) (c) (d) (e)

Figure S16: Cosine similarities between the 10 runs of module extraction of the chosen locality placement and all other locality placement runs, MNIST UMAP embeddings, $o = 12$, $M = 9$.

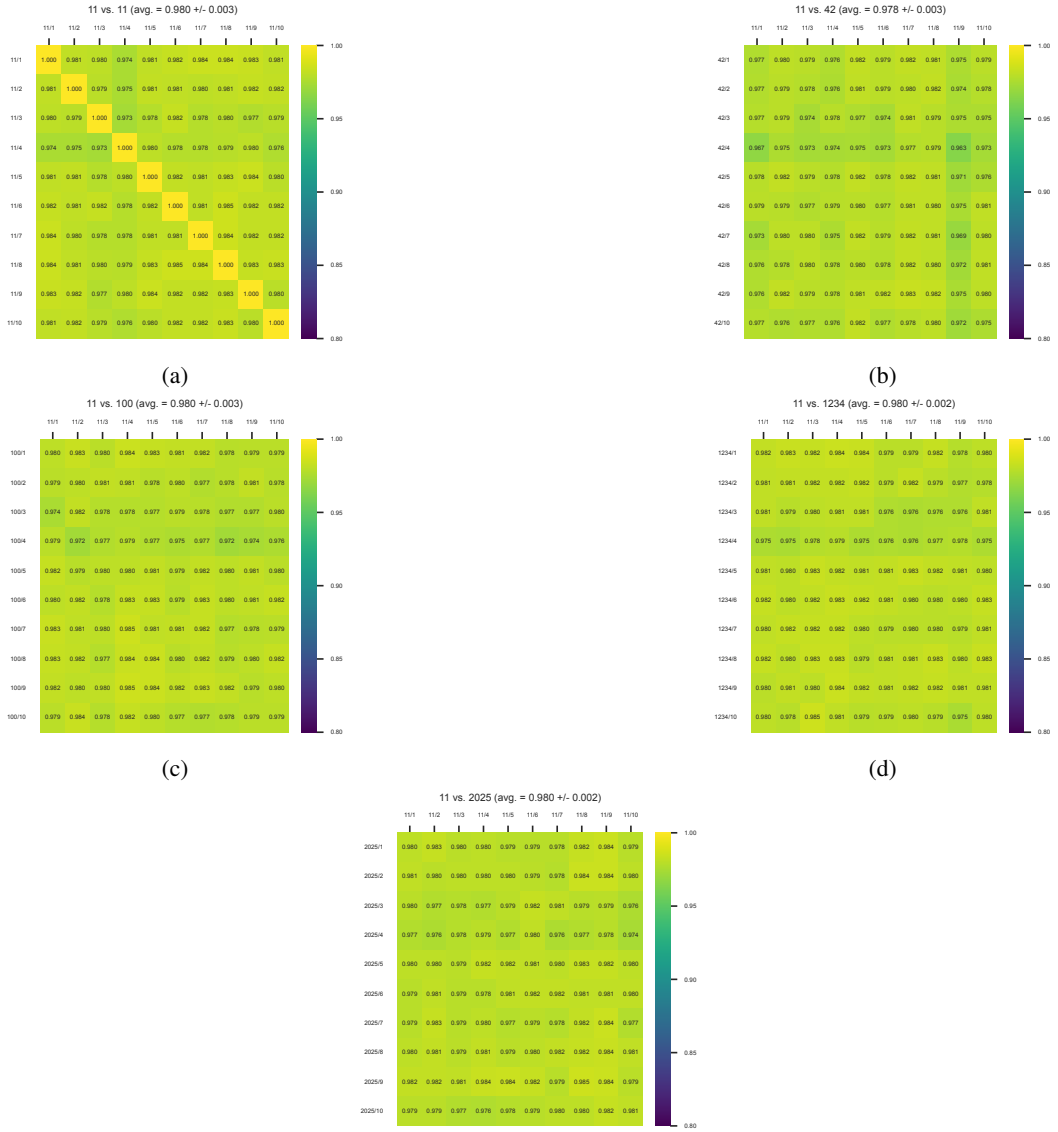
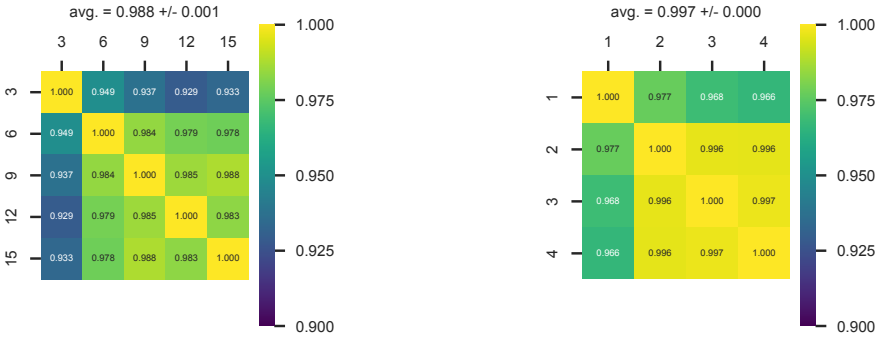


Figure S17: Cosine similarities between the 10 runs of module extraction of the chosen locality placement and all other locality placement runs, KPMP UMAP embeddings, $o = 3$, $M = 9$.



(a) MNIST UMAP embeddings.

(b) KPMP UMAP embeddings.

Figure S18: Cosine similarities between the average presence-weighted module averages across different overlap parameters o.

B.7 MNIST EXPERIMENTS

Here, we show some additional experiments and results done using the MNIST dataset.

B.7.1 UMAP EMBEDDINGS: LOCAL PATTERNS, GLOBAL PATTERNS

With this experiment, we explore the similarity between correlation patterns across a latent space when looking at different subsets of variables, to highlight the utility of extracting modules. **Setup.** We use LAVA locality representations of the MNIST UMAP, without extracted modules. As examples, we select two different localities, and then calculate cosine similarity between their and other localities' respective representations. This is done using correlations for all feature pairs, or using subsets involving only features from the top or bottom half of the images. **Intra- and intercluster heterogeneity.** We observe that patterns of correlations between the original features can be highly dissimilar across localities, even within the same cluster of digits (first column of Fig. S19). This can be conceptualized as a multivariate version of Simpson's paradox, wherein different groupings of data – here LAVA localities – can show different trends in association (Samuels, 1993), and highlights the fact that local feature associations cannot be trivially summarized across an entire dataset. **patterns of correlations alter similarity.** Considering only locality correlations between pixels from the top or bottom half of the images, we observe variations in the cosine similarities across localities (second and third columns of Fig. S19). This indicates how different localities across the latent space might share specific patterns of correlation, but not others, which is precisely what the module extraction in LAVA aims to capture.

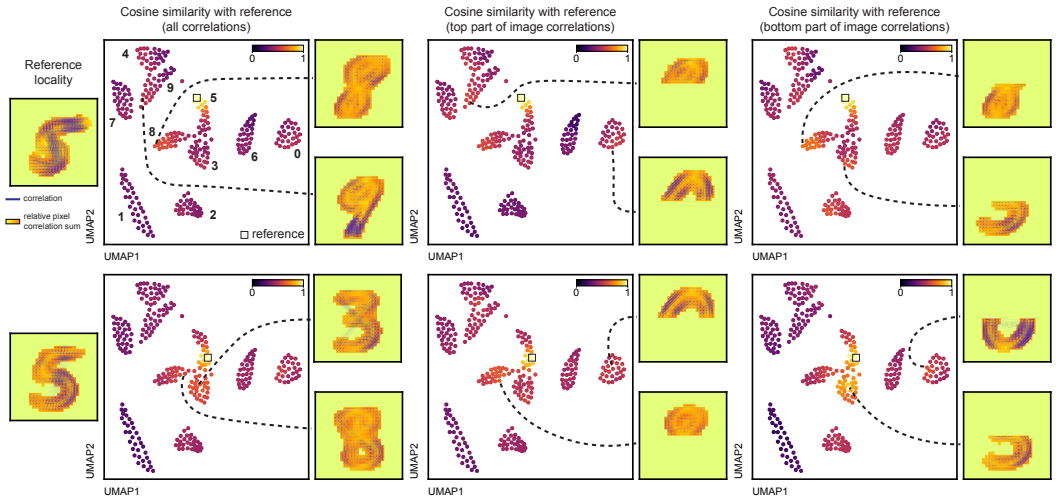


Figure S19: Cosine similarities between each of two reference localities and all other localities of the latent MNIST UMAP space in three settings, with example visualizations of similar localities.

B.7.2 ADDITIONAL RESULTS FOR UMAP EMBEDDINGS

In Fig. S20, we show the class labels of individual samples in our UMAP embeddings. Then, in Fig. S21, we show the relative in-degree centralities of samples when looking at all the neighborhoods and when looking at our locality placement. In Fig. S22, we visualize all modules and their presences used in our experiments from Sec. 3. We also show modules and presences for the second and third best solutions when $L = 9$ (Fig. S23 and Fig. S24).

Finally, in Fig. S25, we also visualize how the chosen modules would look like if they were the presence-weighted average of their localities' correlations, rather than overlapping patterns we extract with AMF (see Sec. 3). While both the presences and modules would change if we were to alter our loss function (they are optimized jointly), the figure still illustrates how the overestimation weighing helps make sure modules we extract are sparse correlation patterns contained in all of the localities the modules are present in, rather than an average correlation across those localities.

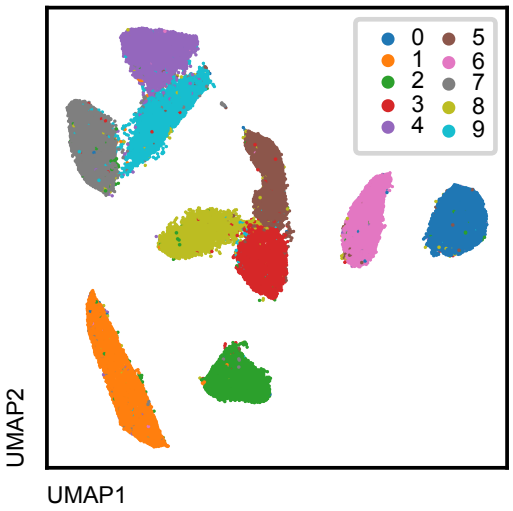
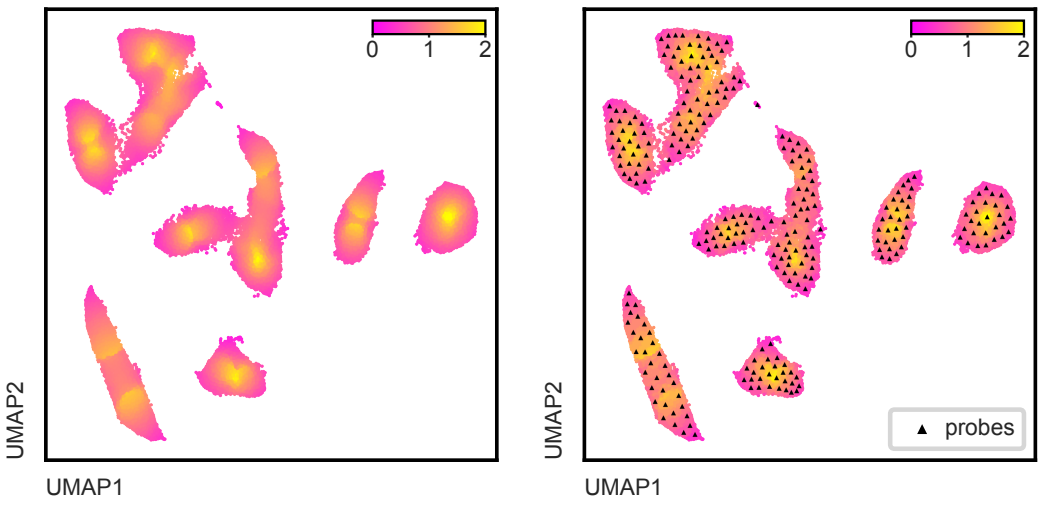


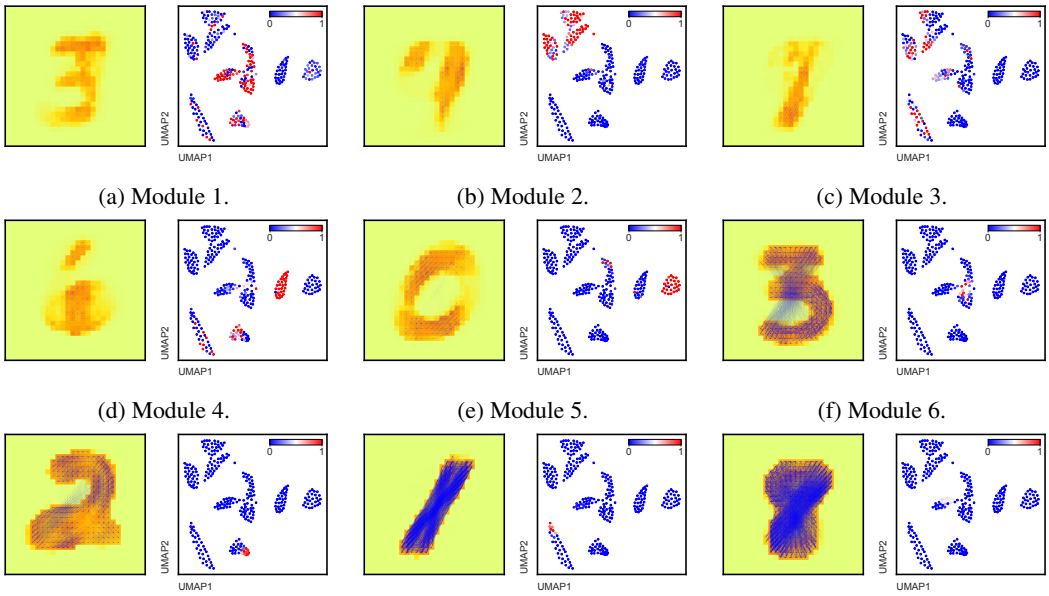
Figure S20: MNIST digit labels across UMAP.



(a) Scaled sample neighborhood in-degree centrality.

(b) Scaled locality in-degree centrality, with probes.

Figure S21: MNIST UMAP sample relative centralities before and after locality placement (scaled by $\frac{E}{n}$ to have an average of 1).

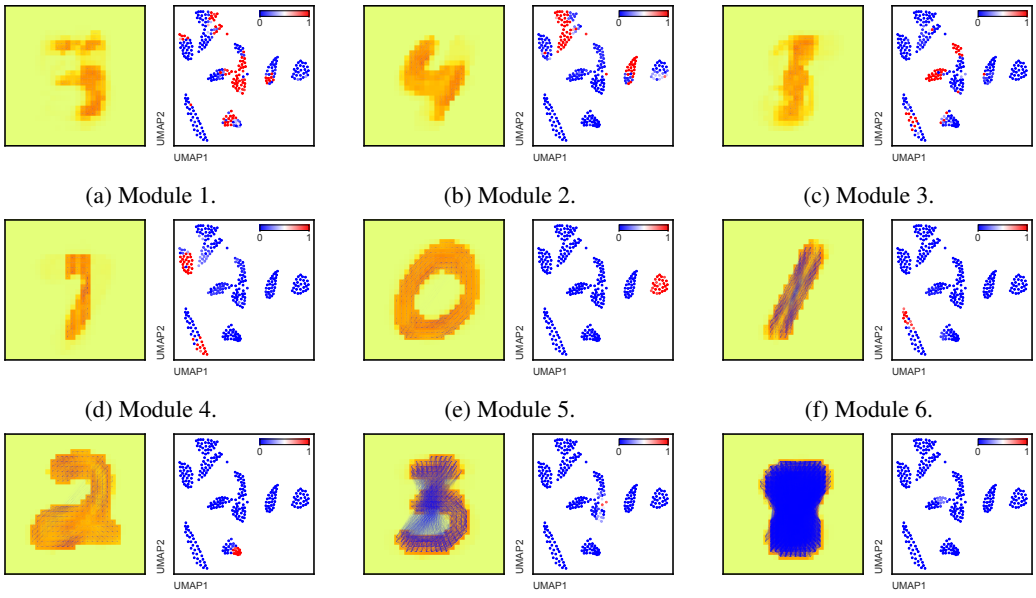


(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S22: Visualizations and presences of the 9 extracted modules for MNIST.

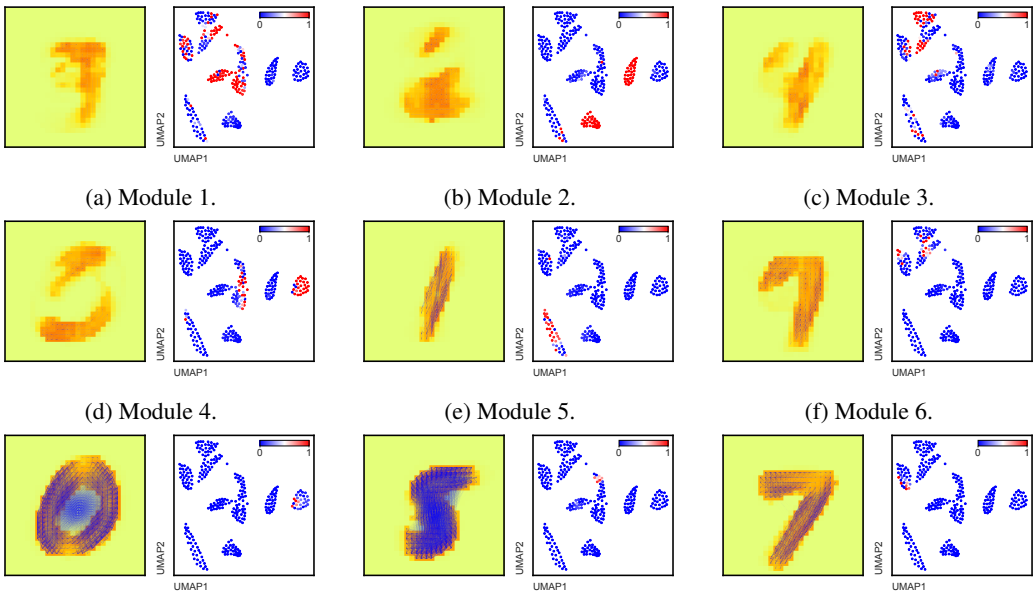


(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S23: Visualizations and presences of the second best 9 modules for MNIST.

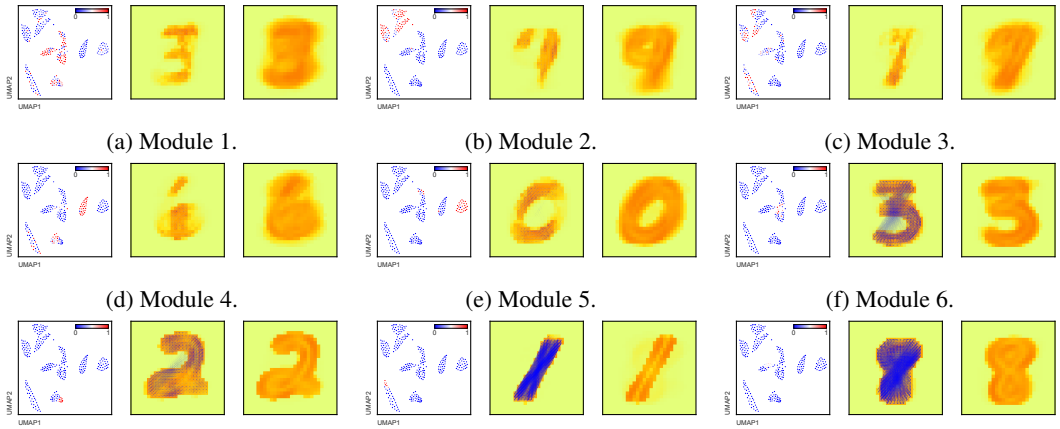


(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S24: Visualizations and presences of the third best 9 modules for MNIST.



(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S25: Presences, actual extracted modules, and what modules would look like if they were presence-weighted averages of the localities they reconstruct, for the 9 extracted MNIST modules.

B.7.3 Additional Results for Autoencoder Embeddings

Here we show some extra results based on the autoencoder embeddings. In visualizations for these embeddings of MNIST, we use a 2-dimensional UMAP (default hyperparameters) of the original 4-dimensional latent space. The UMAP is generated using only the embeddings, and to show a probe location, we use the sample nearest to the probe in the original latent space. Note that all steps of LAVA were run on the original space, not the reduced 2-dimensional UMAP.

In Fig. S26, we show the class label distribution of the latent space. Then, in Fig. S27, we show the relative in-degree centralities of samples when looking at all the neighborhoods and when looking at our locality placement. In Fig. S28 we show the results of running module extraction using a different number modules. Based on these results, we selected the runs with 8 modules, and then used the module extraction run with the lowest main loss term, like in other experiments. In Fig. S29, we visualize all the 8 modules and their presences from the best run, which were used in our LFXAI comparison experiments in Sec. 3.

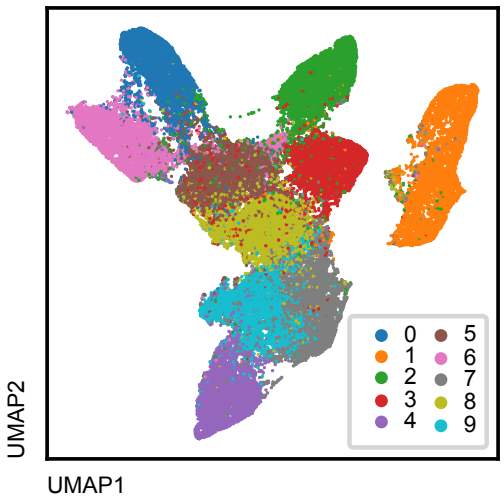
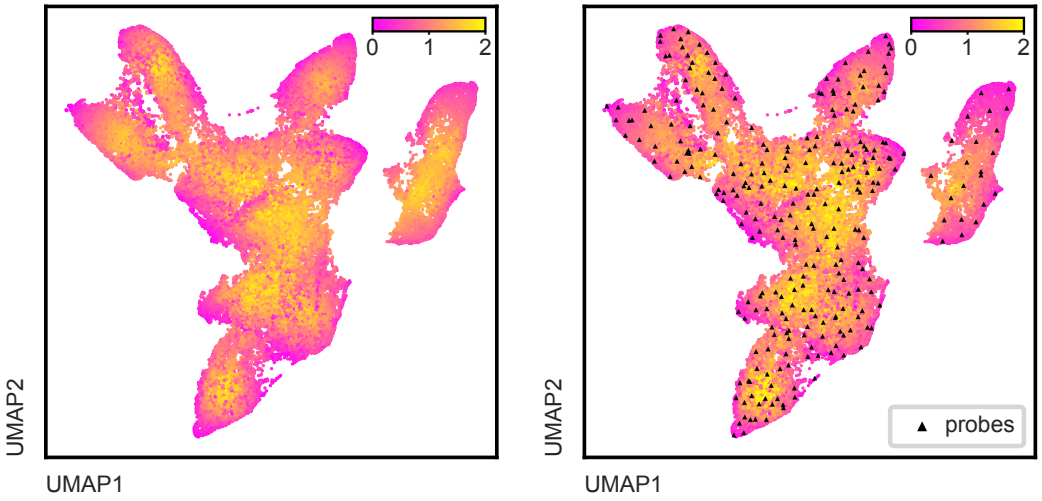


Figure S26: MNIST digit labels across the autoencoder embeddings (UMAP visualization).



(a) Scaled sample neighborhood in-degree centrality.

(b) Scaled locality in-degree centrality, with probes.

Figure S27: MNIST autoencoder sample relative centralities before and after locality placement (scaled by $\frac{E}{n}$ to have an average of 1; visualized using a UMAP of the original embeddings).

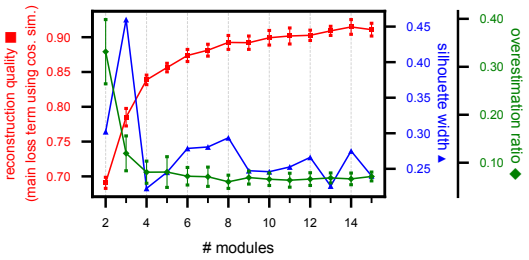
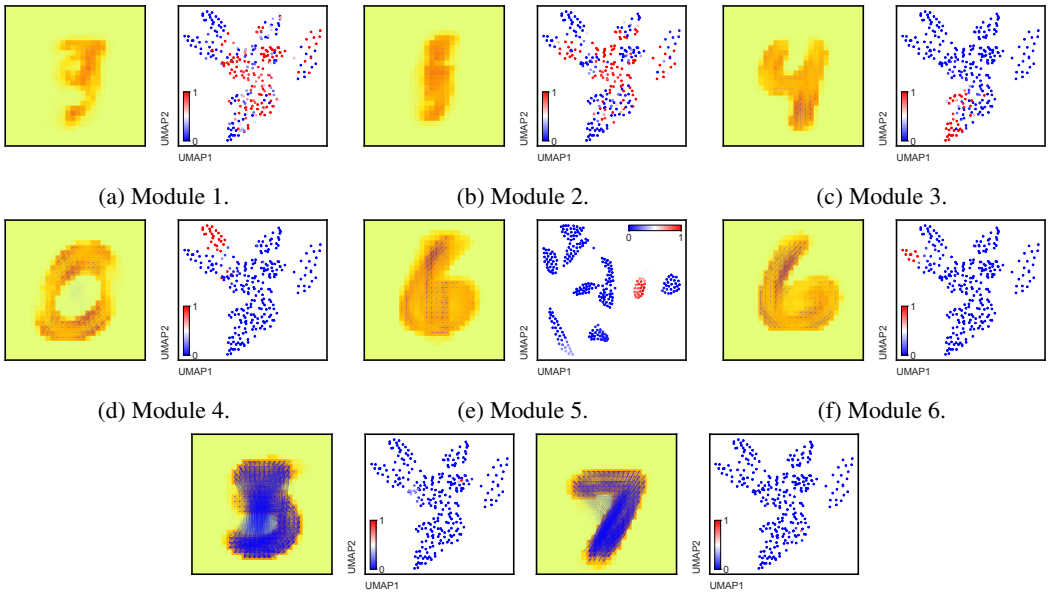


Figure S28: Module extraction statistics for the MNIST autoencoder embeddings.



(a) Module 1. (b) Module 2. (c) Module 3.

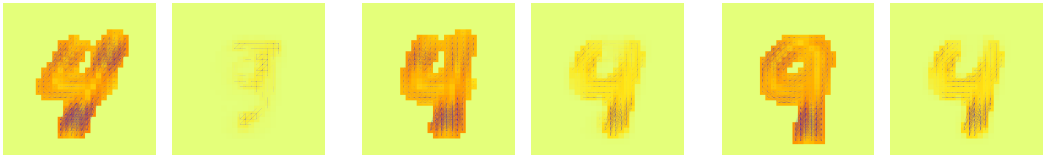
(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8.

Figure S29: Visualizations and presences of the chosen 8 modules for MNIST autoencoder embeddings.

B.7.4 Additional Results for Comparison with LFXAI

Here we expand on Sec. 3.3, and especially Fig. 6. In Fig. S30 we show visualize the correlation patterns of the three example localities alongside their module-based reconstructions. In Fig. S31 we show average LFXAI-based feature importance, alongside their standard deviation (on the same scale as the importance): we can see that standard deviations are generally larger than the importance themselves for these embeddings, making it unclear whether any consistently relevant patterns can be observed in this case. This could be due to the shift equivariant nature of a convolutional autoencoder. We also show a couple of examples of the feature importance before averaging to visualize where the variance comes from (Fig. S32). We additionally note that feature importance used an empty image as a reference (all features were 0, i.e., pixels were white). Another reference could have been used, e.g., random subset of the dataset, or some other reference, but we did not see how altering the reference used could make these explanations more comparable to LAVA without significant alterations.

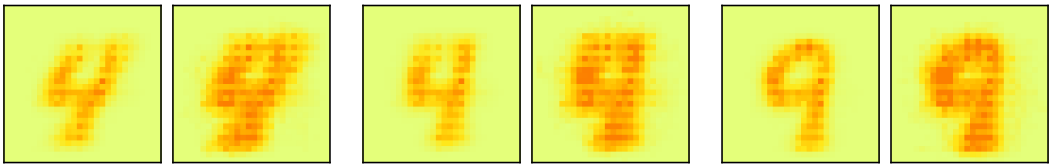


(a) Locality correlations and LAVA reconstruction from example locality 1.

(b) Locality correlations and LAVA reconstruction from example locality 2.

(c) Locality correlations and LAVA reconstruction from example locality 3.

Figure S30: Visualizations of localities used as examples in Fig. 3.3, alongside their reconstructions using LAVA modules.

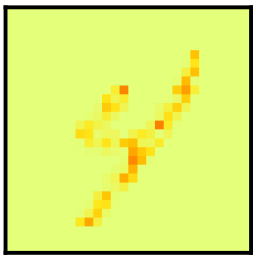


(a) Locality-averaged gradSHAP feature importance and its standard deviation for example locality 1.

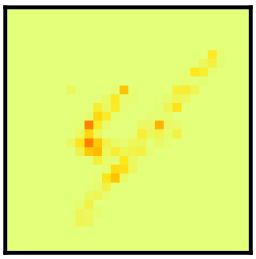
(b) Locality-averaged gradSHAP feature importance and its standard deviation for example locality 2.

(c) Locality-averaged gradSHAP feature importance and its standard deviation for example locality 3.

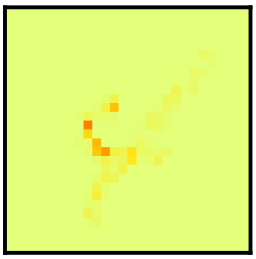
Figure S31: Visualizations of the locality-averaged LFXAI-based gradSHAP feature importance explanations, alongside their standard deviations.



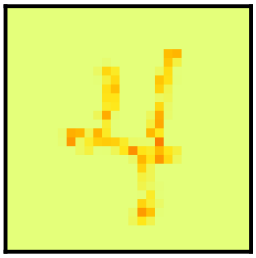
(a) Feature importance of a sample from example locality 1 (1).



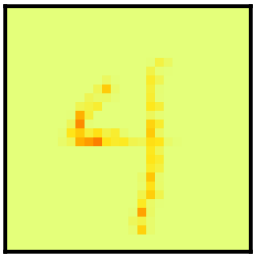
(b) Feature importance of a sample from example locality 1 (2).



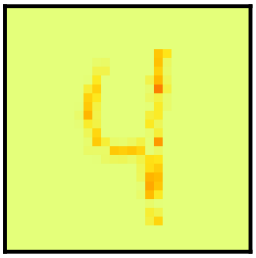
(c) Feature importance of a sample from example locality 1 (3).



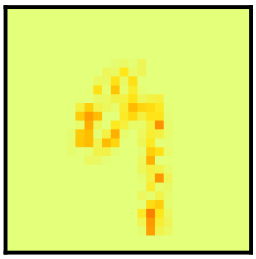
(d) Feature importance of a sample from example locality 2 (1).



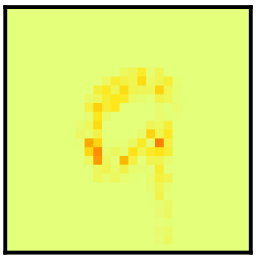
(e) Feature importance of a sample from example locality 2 (2).



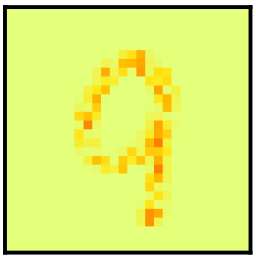
(f) Feature importance of a sample from example locality 2 (3).



(g) Feature importance of a sample from example locality 3 (1).



(h) Feature importance of a sample from example locality 3 (2).



(i) Feature importance of a sample from example locality 3 (3).

Figure S32: Visualizations of LFXAI-based gradSHAP feature importance for the three example localities used in Fig. 6.

B.8 KPMP Experiments

Here, we show some extra figures and results relating to experiments done with the KPMP dataset, and we provide more details on the gene set analysis done in Sec. 3.4.

B.8.1 Additional figures and results.

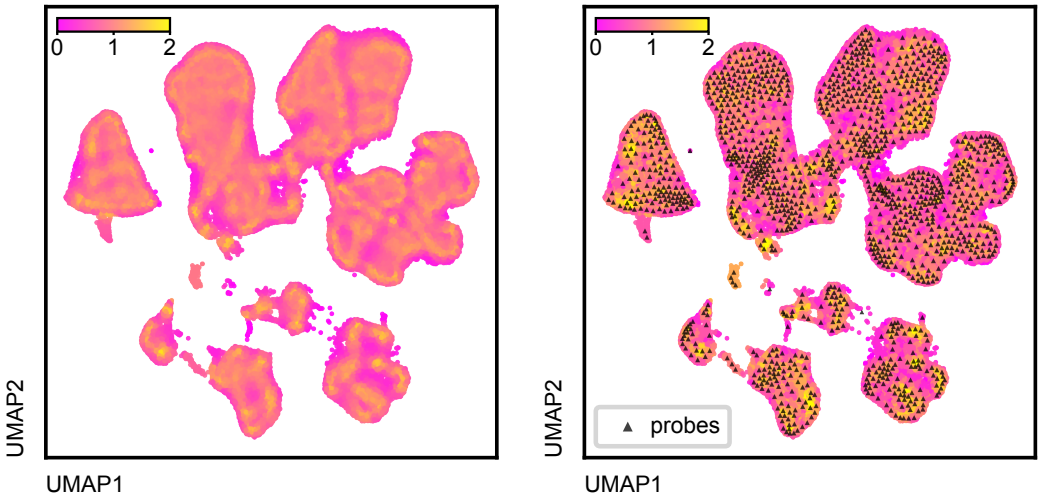
In Fig. S33, we show cell type and disease category labels for all the samples in the UMAP embeddings. In Fig. S34, we show the relative in-degree centralities of samples when looking at all the neighborhoods and when looking at our locality placement. In Fig. S35, we show the presences of modules chosen for experiments (Sec. 3.4).



(a) Cell types.

(b) Disease states.

Figure S33: KPMP metadata.

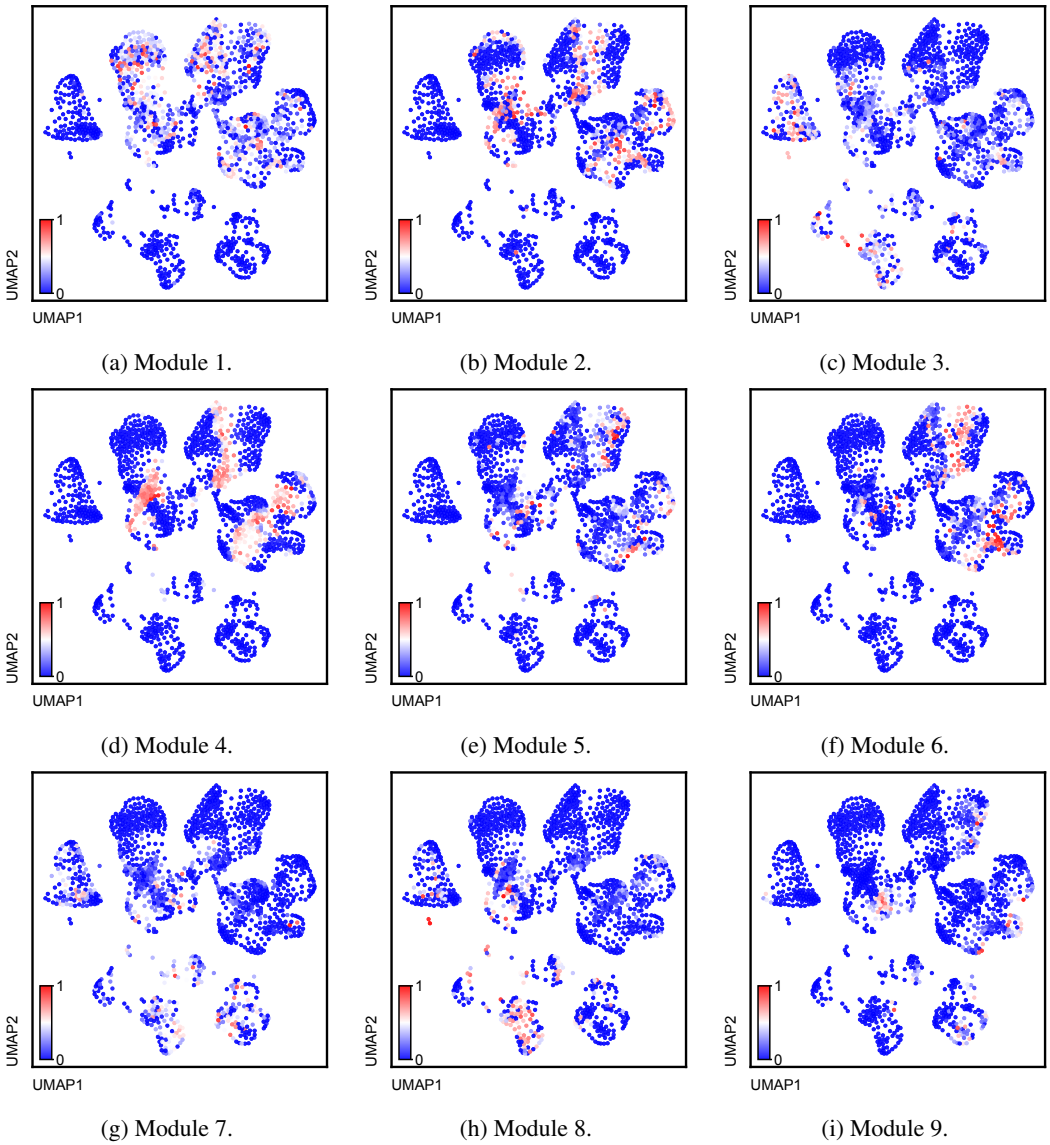


(a) Scaled sample neighborhood in-degree centrality.

(b) Scaled locality in-degree centrality, with probes.

Figure S34: KPMP UMAP relative sample centralities before and after locality placement (scaled by $\frac{E}{n}$ to have an average of 1).

Disease category. In Sec. 3.4, we analyze our modules in relation to disease category of patients from whom the single-cell samples come from. To get a better sense of the data, in Fig. S36, we visualize the ratios of diseased samples across LAVA's localities. **Correlations between module presences and disease category ratios.** For each of the modules, we also report correlations between the presences of that module and the ratios of the three different disease categories (Table S2). We calculate correlations only across localities in which the module is actually present, that is, with presence > 0.01. We do this because we do not expect that a pattern of correlations in a locality will



(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

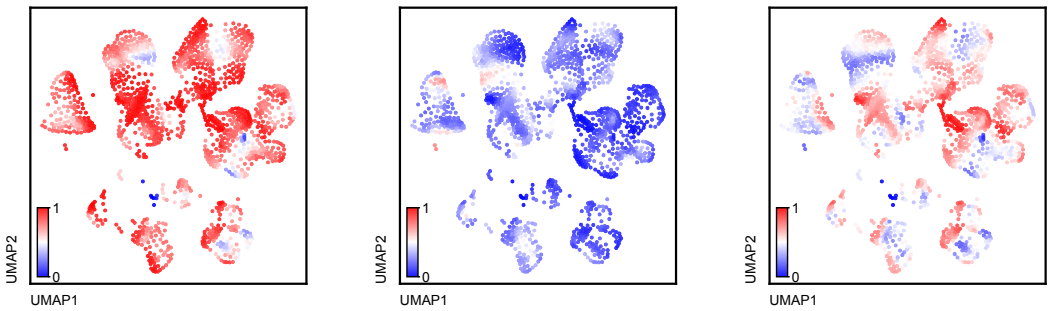
(g) Module 7. (h) Module 8. (i) Module 9.

Figure S35: Presences of the 9 extracted modules for KPMP.

be caused purely by the number of disease samples. Therefore, we also do not expect any module presences to directly reflect the number of disease samples across the localities. However, we do expect that the presence of a module might reflect the relative number of disease samples across the localities it is present in, as a disease could have a specific influence on gene expression for specific cells, and hence also the correlations we might observe when looking at the embeddings locally.

	CKD + AKF, Pear.	CKD + AKF, Spear.	CKD, Pear.	CKD, Spear.	AKF, Pear.	AKF, Spear.
m. 1	$-.06$ ($p = .116$)	$-.06$ ($p = .139$)	$-.11$ ($p = .005$)	$-.08$ ($p = .035$)	$.07$ ($p = .049$)	$.07$ ($p = .052$)
m. 2	$.01$ ($p = .865$)	$.03$ ($p = .473$)	$-.03$ ($p = .592$)	$-.02.$ ($p = .62$)	$.05$ ($p = .292$)	$.08$ ($p = .095$)
m. 3	$-.01$ ($p = .878$)	$-.05$ ($p = .242$)	$-.18$ ($p < .001$)	$-.20$ ($p =$)	$.23$ ($p < .001$)	$.25$ ($p < .001$)
m. 4	$.23$ ($p < .001$)	$.26$ ($p < .001$)	$.28$ ($p < .001$)	$.28$ ($p < .001$)	$-.05$ ($p = .269$)	$.03$ ($p = .501$)
m. 5	$-.05$ ($p = 219$)	$-.08$ ($p = .092$)	$-.11$ ($p = .010$)	$-.14$ ($p = .002$)	$.09$ ($p = .042$)	$.08$ ($p = .065$)
m. 6	$-.08$ ($p = .107$)	$-.07$ ($p = .173$)	$-.13$ ($p = .010$)	$-.13$ ($p = .013$)	$.08$ ($p = .126$)	$.12$ ($p = .014$)
m. 7	$-.31$ ($p < .001$)	$-.36$ ($p < .001$)	$-.28$ ($p < .001$)	$.28$ ($p < .001$)	$-.01$ ($p = .846$)	$-.01$ ($p = .790$)
m. 8	$-.13$ ($p = .008$)	$-.17$ ($p < .001$)	$-.22$ ($p < .001$)	$-.20$ ($p < .001$)	$.14$ ($p = .003$)	$.11$ ($p = .028$)
m. 9	$.10$ ($p = .084$)	$.09$ ($p = .127$)	$.04$ ($p = .429$)	$.06$ ($p = .310$)	$.09$ ($p = .099$)	$.14$ ($p = .015$)

Table S2: Correlations between module presences and disease categories.



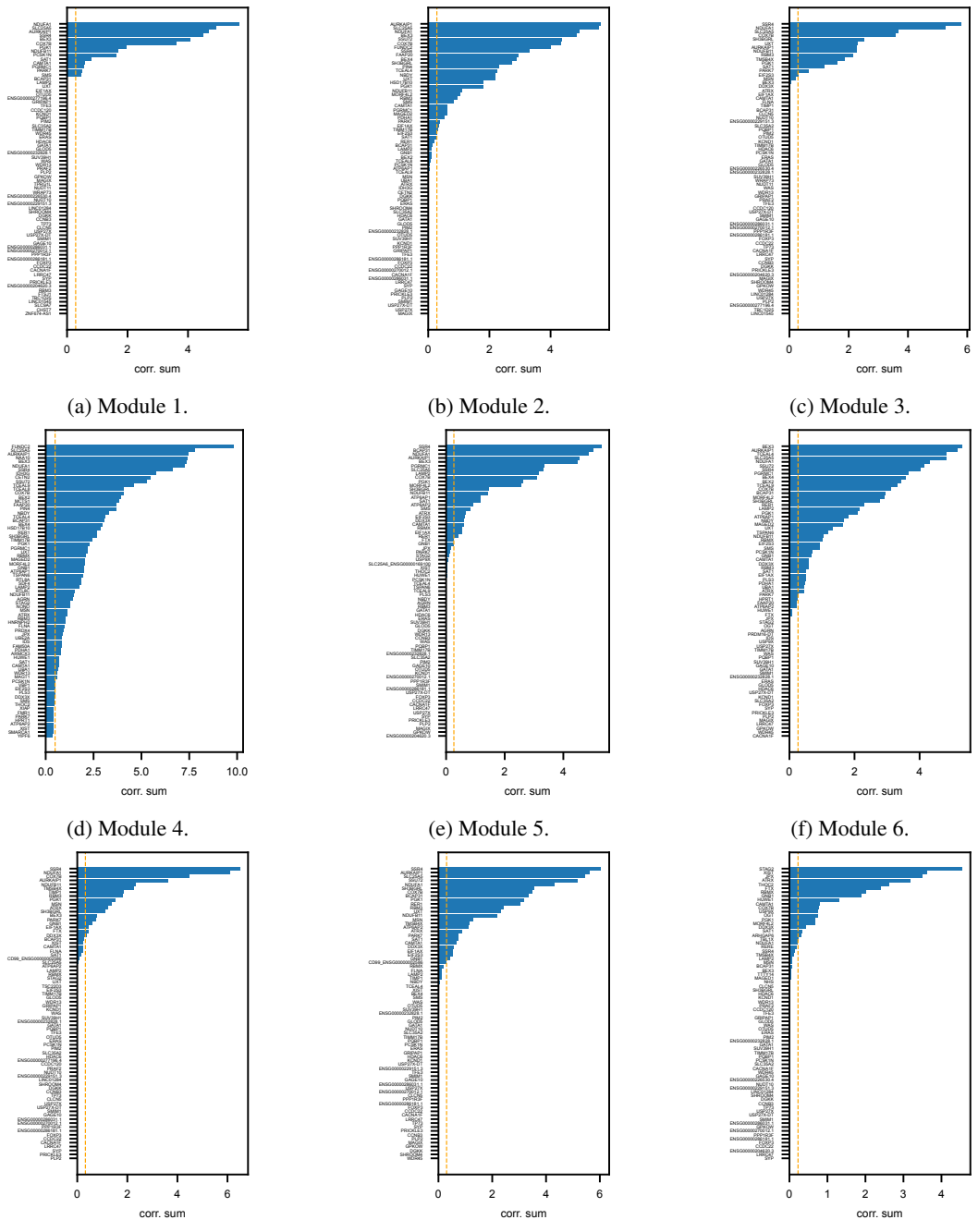
(a) Any disease. (b) AKF: acute kidney failure. (c) CKD: chronic kidney disease.

Figure S36: Percentage of disease categories across localities.

B.8.2 Gene pathway analysis

In Sec. 3.4, we investigate the biological relevance of the patterns uncovered by LAVA based on the UMAP of the KPMP dataset. To do this, we perform overrepresentation analysis (ORA) of biological functions: a statistical method used to determine which functions are statistically overrepresented amongst the functional annotations of the genes in a set of interest (e.g. a LAVA module), compared to what would be expected by chance (Huang et al., 2009). We perform this analysis using three collections of functional annotations we thought were relevant for our disease context: (1) hallmark; (2) canonical pathways; and (3) the gene ontology biological process. These collections were downloaded from MSigDB [15]. **Selection of genes per module.** To select the set of genes of interest for ORA, we first sum the correlations of each gene in a module, and then take the set of genes whose sum of correlations is $\geq 5\%$ of the largest sum (Fig. S37). Accordingly, we performed ORA for each module using as background all measured genes from the KPMP dataset (a total of 31332 genes). The resulting biological function p-values were adjusted using the Benjamini-Hochberg correction for multiple testing. **Results of ORA.** We report the top 50 most significant results of ORA for each of the 9 extracted modules in Fig. S38. In the main paper, we highlight module 4 because of its correlation with diseased samples, and CKD specifically (Sec. 3.4, Table S2), which we think is of biological interest.

[15] The Molecular Signatures Database, available at https://www.gsea-msigdb.org.

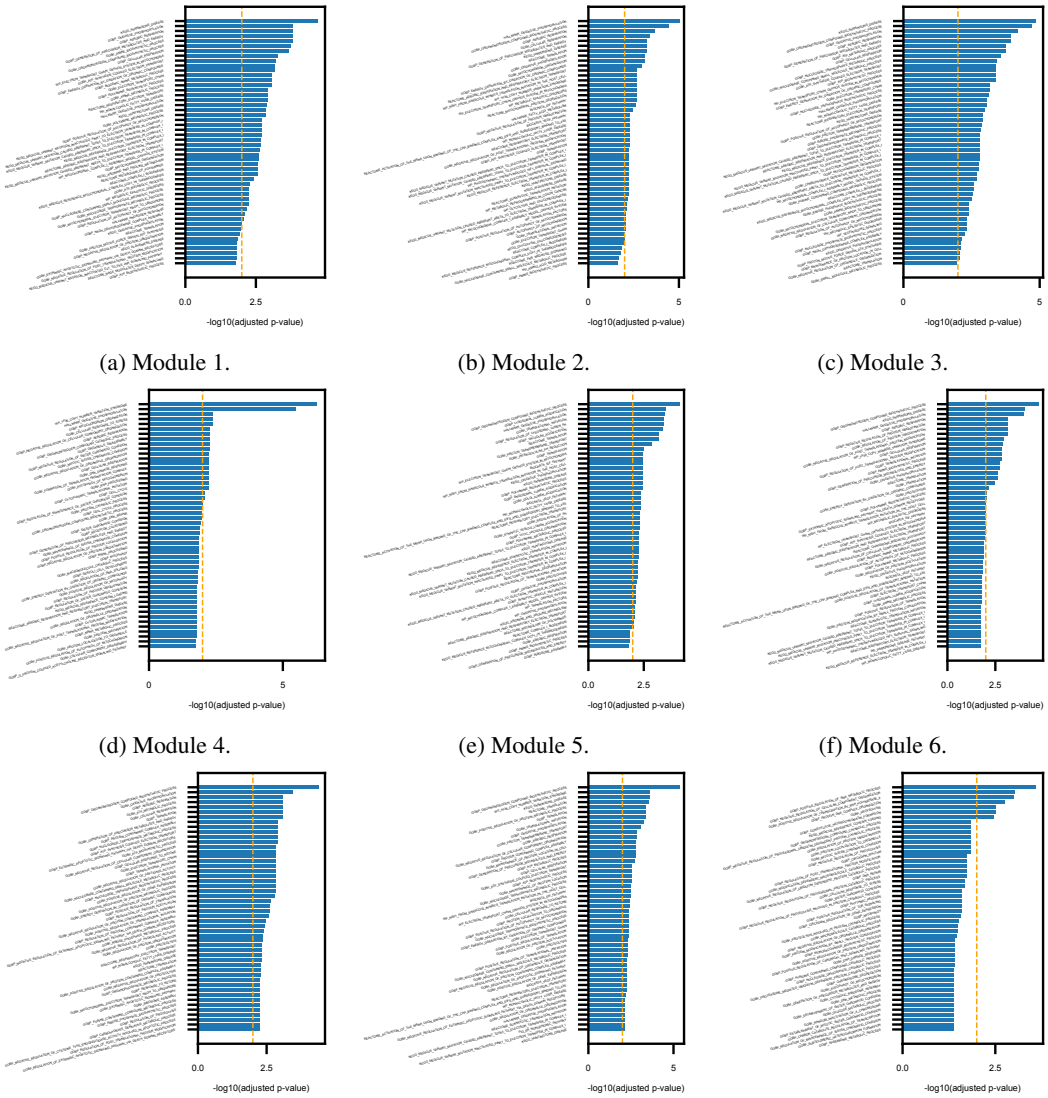


(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S37: Top 75 genes based on sums of correlations for each extracted module from the KPMP experiments. The orange line represents the cut-off line for gene subsets for ORA, equal to 5% of the largest correlation sum.



(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S38: Top 50 most significant pathways based on ORA for each of the 9 extracted modules from the KPMP experiment. The orange line represents the significance p-value threshold of 0.01, i.e., 2 when looking at $-log_{10}(p)$ values.

B.9 BOW EXPERIMENTS

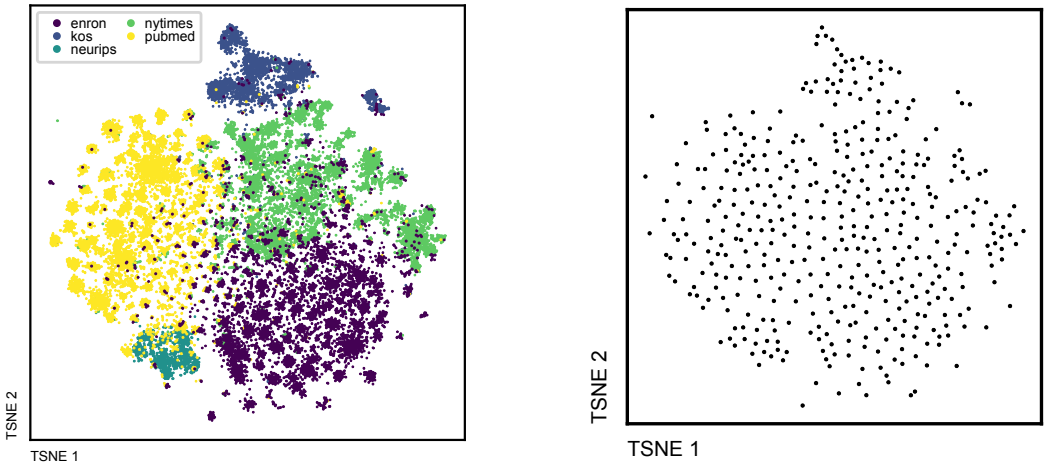
To explore whether LAVA can be applied to natural language processing (NLP) settings, in addition to the MNIST and KPMP datasets, we run LAVA on embeddings of the Bag-of-Words (BOW) dataset Newman (2008). The BOW dataset originally contains 8000000 of documents from 5 different corpora: NeuRIPS articles, Enron emails, KOS blog entries, PubMed abstracts, and New York Times articles. We subsample the dataset, generate semantic embeddings, and then extract LAVA explanations.

B.9.1 PREPROCESSING AND EMBEDDING

First, we randomly subsample 10000 documents from each of the 5 text corpora. In case of corpora with less than 10000 documents (KOS and NeuRIPS with 3430 and 1500 documents, respectively), we use all documents. This yields a total of 34930 documents.

Second, to embed the sub-sampled BOW dataset, we run the scikit-learn TF-IDF implementation with default hyperparameters and maximum and minimum document frequencies at 0.75 and 0.01, respectively, to obtain a 300-dimensional representation of the data. Finally, based on the 300 TF-IDF dimensions, we generate a t-SNE embedding using cosine distance and default hyperparameters (Fig. S39a). We observe that the corpora are separated clearly from one another, with some overlap between them, such as NeuRIPS and PubMed, which we could expect due to overlap of topics and use of language. However, the structure of the embeddings exhibit a lot of smaller clustering, which could be a reflection of either more specific subtopic of the documents, or parameters of the t-SNE.

As features for LAVA, we use the 1000 most variably occurring words. We then normalize frequencies of those 1000 words in any document to sum up to 1, which enables locality representation to measure whether words frequencies in localities increase or decrease together in relative terms. Locality representation should in that case relate to the prevalence of some topic in documents of a locality.



(a) t-SNE embeddings of the TF-IDF preprocessed BOW dataset.

(b) LAVA probe placement for $n = 500$ and $o = 6$.

Figure S39: t-SNE embeddings and LAVA probes for the BOW dataset.

B.9.2 LAVA EXPLANATIONS

Setup. For localities, we choose $n = 500$ (following the same reasoning as for KPMP in App. B.4.1), and $o = 6$ (Fig. S39b). For module extraction, we use the default $\nu = 9$ hyperparameter. Similarly to the KPMP dataset, we notice lower values of the γ hyperparameter have no effect on module extraction (overestimation ratios are not controlled), so we increase the value to $\gamma = 150$, where the impact becomes visible. As in other experiments, we run AMF $R = 10$ times with different numbers of modules and report statistics (Fig. S40).

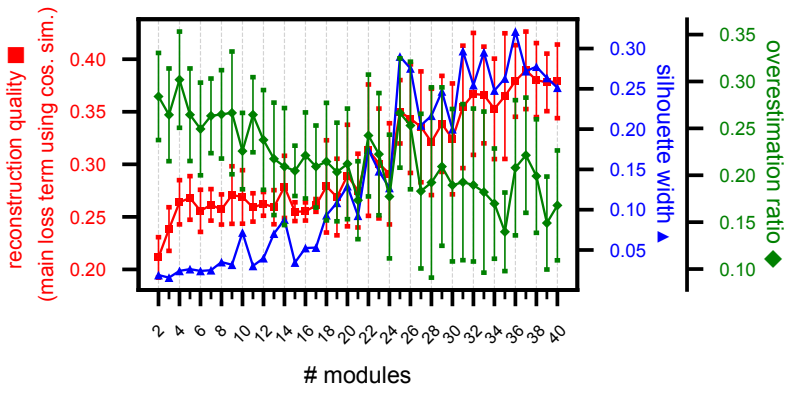


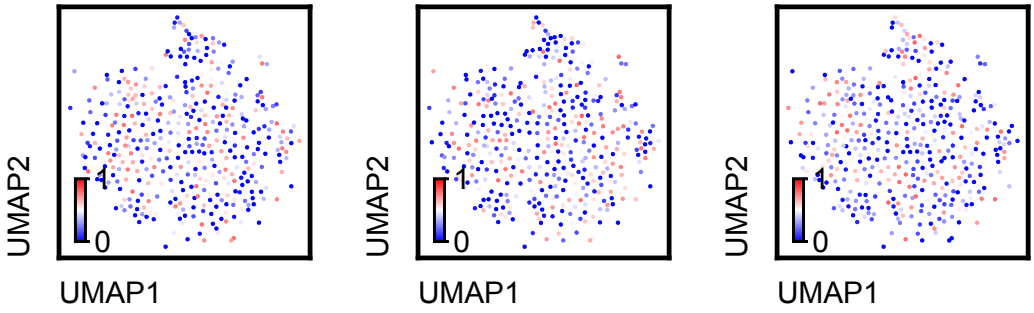
Figure S40: Module extraction statistics for the t-SNE BOW embeddings.

Results.

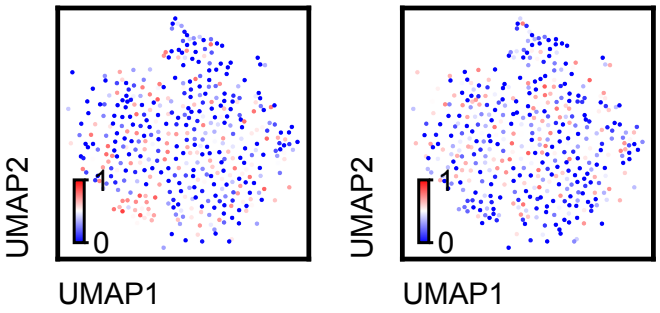
First, we see that AMF fails to extract lower number of modules with high reconstruction qualities. This could indicate that localities exhibit heterogeneous correlation patterns, which cannot be accurately reconstructed using a smaller number of modules. Second, we see very low silhouette widths when M is small. This, together with relatively stable reconstruction qualities of the different runs, indicates there are many module factorizations of similar quality, further reinforcing the fact that localities are too heterogeneous. This is further reinforced by the visible increase in both stability and reconstruction quality as M increases. Third, we observe overestimation ratios to be higher than the 0.10 we would expect for $\nu = 9$ (Apps. A.4 and B.4.2). As smaller γ values have no impact on the the loss (see setup above), this might indicate that there is a tradeoff between the main loss term and the scale regularization term. This explanation strengthened by the fact that increasing M increases reconstruction quality while decreasing overestimation, as in these cases AMF is not forced to sacrifice scale for additional flexibility.

We also observe correlations and presences of modules when $M = 5$, like in the KPMP experiments (Figs. S41-S43). We see that the 5 modules extract a more general pattern of word correlation (module 1), several patterns relating to politics (modules 2, 3, and 5; containing words such as government, campaign, and president) and a pattern relating to scientific terminology (module 4; containing correlations with words such as number, point, and analysis). Notably, the scientific pattern is highly present in localities of NeuRIPS articles, which corresponds to our expectations of the semantic content of those documents. Multiple politics-related modules capturing similar words, but different amount of correlations, confirms the heterogeneity of localities not only at the level of correlations and words present, but also their relative amounts of these correlations. This is to be expected due to sparsity of word frequencies, causing relatively noisy/varying amounts of correlations across the localities, even if their documents are related to semantically. Heterogeneity is also visible in how spread-out the modules presences are.

In summary, LAVA is successful at uncovering reoccurring topical patterns that are a product of the latent structure, however, the correlation patterns of relative word frequencies of might in general be too noisy and heterogeneous to yield more meaningful results for semantic analysis. A way to generate more relevant LAVA explanations could be to extract higher level features. For example, we could group larger numbers of words based on similar meaning, which then might show less noisy and more consistent and less overlapping modules. Stretching representation further, if embeddings were based on an attention-based model, we could look at attention the model places on words rather than their frequency as features. We expand on other possibilities of extending the LAVA framework with different types of locality representation more generally in App. A.3.

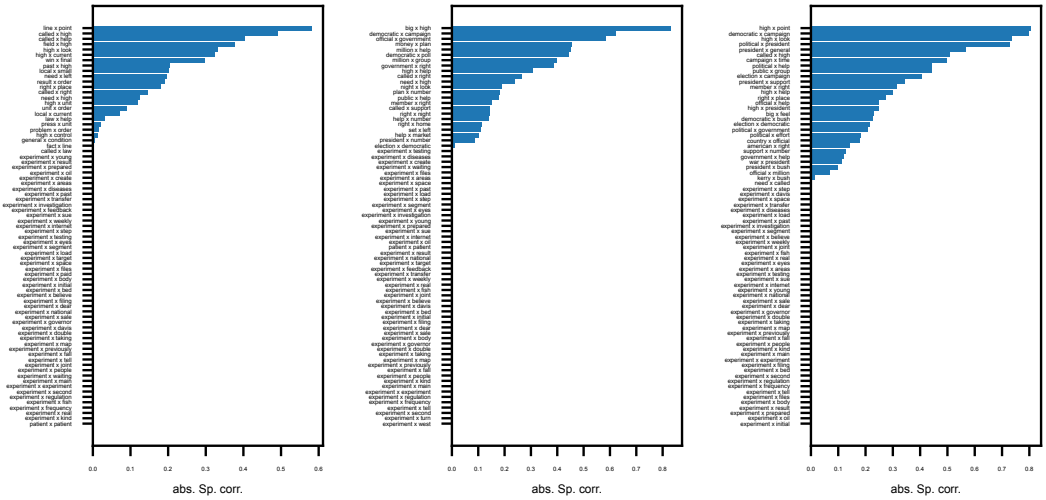


(a) Module 1. (b) Module 2. (c) Module 3.

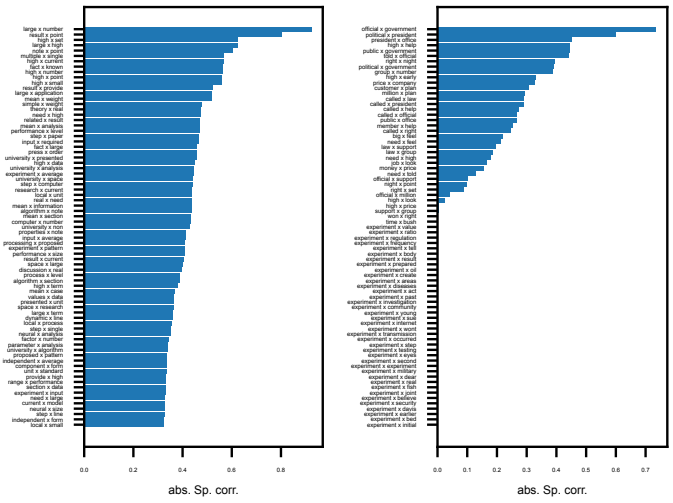


(d) Module 4. (e) Module 5.

Figure S41: Presences of LAVA modules for the t-SNE/TF-IDF embeddings of the BOW dataset, $M = 5$.

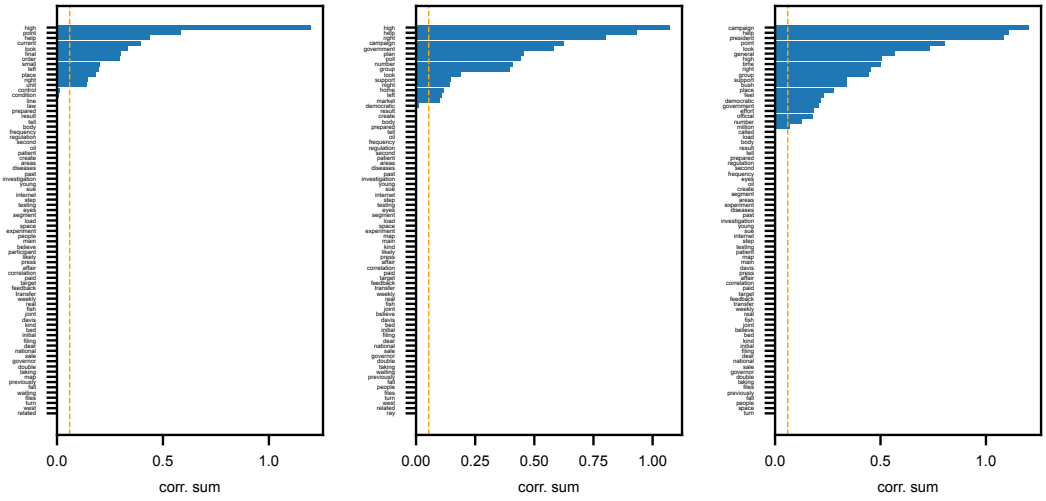


(a) Module 1. (b) Module 2. (c) Module 3.



(d) Module 4. (e) Module 5.

Figure S42: The top 75 correlations of LAVA modules for the t-SNE/TF-IDF embeddings of the BOW dataset, $M = 5$.



(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5.

Figure S43: The top 50 features of LAVA modules for the t-SNE/TF-IDF embeddings of the BOW dataset, $M = 5$.

B.10 COMPARING AMF TO NMF

Here we compare AMF to Non-negative Matrix Factorization (NMF), the method which we found to be conceptually most similar, and therefore most relevant to motivate the need for developing a new method for module extraction (Lee & Seung, 1999).

The aim of step 3 in LAVA is to capture patterns of correlation across latent embeddings, as defined through localities. While in principle it might appear suitable to use a more general factorization approach, AMF is designed with the following additional properties in mind: to capture reoccurring rather than averaged patterns of correlation across latent embeddings, to focus on patterns rather than absolute values, and to focus on localities with many high correlations as opposed to few or mostly weak correlations. These aims led to the corresponding modeling choices for AMF, motivated in Sec. 2, which also make it different from other matrix factorization methods.

B.10.1 CONCEPTUAL DIFFERENCES BETWEEN AMF AND NMF

NMF decomposes some non-negative matrix V $(n \times m)$ into non-negative matrices W $(n \times k)$ and H $(k \times m)$, where k is a hyperparameter that determines the number of "components" or "parts", and W and H are optimized so that $V = WH$[16]. While NMF and AMF are both "parts-based" matrix decompositions, we highlight three conceptually distinct features of AMF: (1) entries of the AMF module and presence matrices are restricted to real values in the $[0, 1]$, whereas in NMF matrices are restricted only to non-negative values; (2) the AMF method uses outer products and a maximum function to reconstruct the matrix (localities with their corresponding correlations), rather than standard matrix multiplication like in NMF (Sec. 2.3; Figs. 2 and 5); (3) AMF uses a similarity-based loss function that incorporates a quantile-regression-like weighing scheme, rather than optimizing some form of beta-divergence as is the case with NMF. We motivate these essential modeling choices throughout Sec. 2.3 in terms of the desired properties and interpretability of the modules, and we showcase the effectiveness of our formulation in terms of module (and presence) properties in Sec. 3.2. We stress again that both the formulation and loss function of AMF allow us to interpret modules meaningfully, while NMF remains incoherent for factorization of correlation.

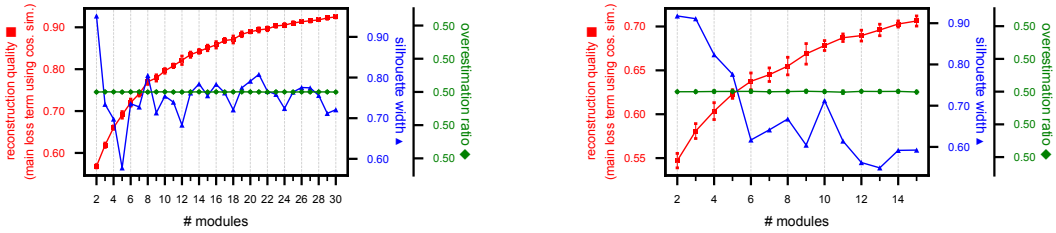
B.10.2 EMPIRICAL RESULTS OF USING NMF TO EXTRACT MODULES

To illustrate some of the differences of the parts/modules extracted by the two methods, we apply NMF to the LAVA localities of the UMAP embeddings of MNIST and KPMP (from our main paper experiments). For NMF, we use random initialization and the KL divergence as the loss for NMF. We deem KL divergence to be the beta-divergence that is most similar to our cosine distance loss function.

First, we report module extraction statistics when varying the number of modules/components in Fig. S44 (we run NMF 10 for each number of components, just like in Fig. 3b). The first thing we note is that the overestimation ratio is consistently 0.5, which means that if we look at NMF components that are used to reconstruct the various localities, on average half of the correlations in those parts will be overestimated and half underestimated, which makes it impossible to treat them as parts of the underlying correlations of localities, but rather as their averages (which is one of the things we prevent with our AMF formulation). We also notice that optimizing KL divergence is indeed similar to our cosine-distance-based loss, as the loss function inverse increases steadily with the increase of number of NMF parts (referred to as modules in the figure). However, NMF does yield higher silhouette widths, indicating that its formulation will produce more stable solutions. We already comment on why we think this is the case with AMF in Sec. 3.2. To further quantify the differences between the AMF modules and NMF components, we calculate the Shannon entropy of every extracted module for all of the module extraction runs above, and compare them between AMF and NMF (Fig. S45). A higher entropy indicates that the correlation pattern is more dense, while a lower entropy means that the pattern is sparser. We can again see that NMF will consistently produce more dense components, coinciding with the fact that it does not optimize for lower overestimation errors, and is therefore more likely to average localities rather than extract their sparser, reoccurring subparts. We also calculate entropy of the presences and coefficients (for AMF and NMF, respectively; Fig. S46). Here, a higher entropy means a module (or

[16]Using standard NMF naming convention for the different variables, not to be confused with any of the variable names from our method.

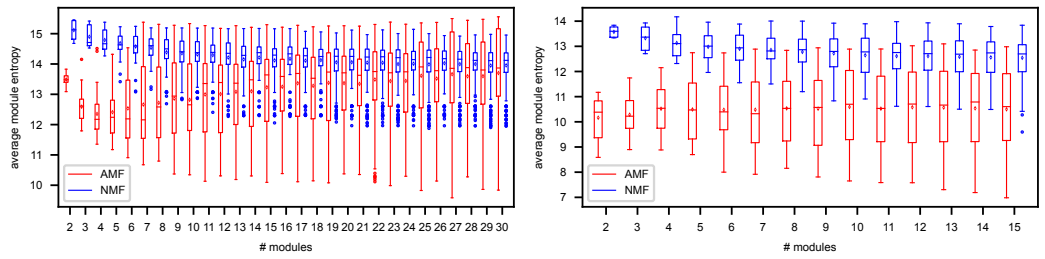
component) is more spread-out across the localities (more global), and vice versa. As expected from the observed module entropy, we see that AMF modules also differ in terms of how spread they are, with less spread modules likely corresponding to higher entropy values (more dense and local patterns). When putting the dynamics of both entropy values together, we see that AMF modules will start with sparser and more spread-out modules than NMF, and continue to produce both more spread-out and sparser modules as we increase their number. We can further infer that when using higher numbers of modules, the correlation patterns in our embeddings will produce some sparse but widely-reoccurring AMF modules (low entropy modules with high entropy presences), while the rest of the modules will be spent on capturing the correlations patterns of the more specific localities that do not contain these sparser patterns (high entropy modules with low entropy presence), as could be expected from the loss function. When looking at NMF in relation, we can see that it will produce less sparse and less spread-out components from the start, with some sparser components occurring more often only when we increase the number of components (still not as sparse as AMF). Finally, we also confirm this by showing a selections of visualizations of NMF-produced components. We use a run with 9 modules whose reconstructions had the lowest AMF loss to enable a fair qualitative comparison with our AMF modules from Sec. 3 (Fig. S47). The visualizations mainly confirm what we already observe quantitatively: while the more dense modules do manage to capture some wide-spread patterns of correlations, the fact that we do not control overestimation (e.g., module 5 that clearly averages across the correlation of localities containing images of the numbers 1 and 6) and the fact that NMF adds up components rather than overlaps them like AMF (Fig. 5c) makes interpretation of the NMF-captured correlation patterns virtually impossible.



(a) MNIST UMAP NMF module extraction.

(b) KPMP UMAP NMF module extraction.

Figure S44: Module extraction runs (10 runs per $\#modules$) using NMF for the UMAP embeddings of the MNIST and KPMP datasets.

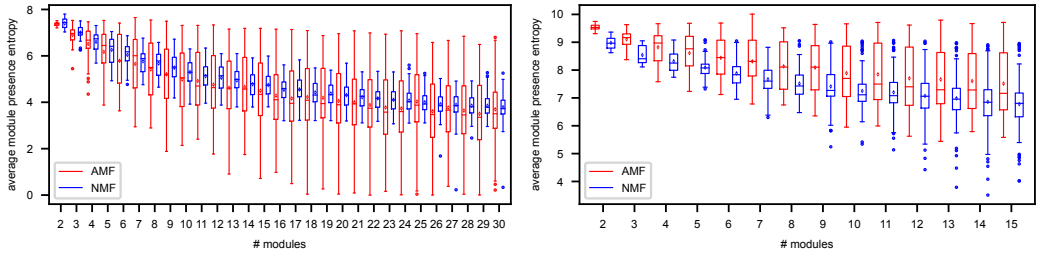


(a) MNIST UMAP module entropy.

(b) KPMP UMAP module entropy.

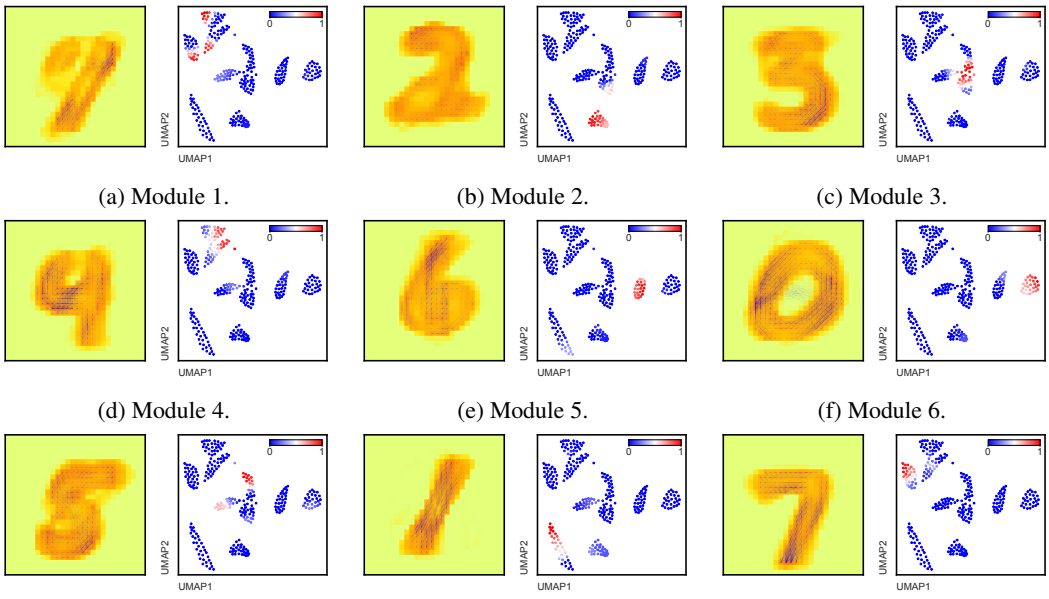
Figure S45: Shannon entropy of modules across the module extraction runs (10 module runs per $\#\ modules$).

There are other variants of NMFs, such as NMF with sparse constraints (Hoyer, 2004), which could produce lower entropy modules. However, we do not take it upon ourselves to perform additional experiments to evaluate these properties, as the conceptual differences and lack of overestimation guarantees we highlight still make any of these alternatives unfit for explaining the variation of correlation patterns in the latent localities. Also outside of the scope of this work, it might be interesting to evaluate whether there are theoretical or empirical similarities between sparsity and overestimation constraints, and whether an overestimation-constrained NMF might yield more interpretable factorizations in specific domains (e.g., mutational signature detection).



(a) MNIST UMAP module presence entropy. (b) KPMP UMAP module presence entropy.

Figure S46: Shannon entropy of module presences across the module extraction runs (10 module runs per $\#\ modules$).



(a) Module 1. (b) Module 2. (c) Module 3.

(d) Module 4. (e) Module 5. (f) Module 6.

(g) Module 7. (h) Module 8. (i) Module 9.

Figure S47: Visualizations of the NMF components and their coefficients across the LAVA-generated localities.

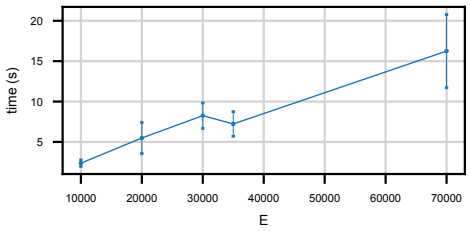
B.11 Time Complexity and Runtime Experiments

LAVA is a scalable framework with parallelized components. In this section we examine time complexity and perform runtime experiments for each of the three steps of LAVA. Runtime experiments were run locally, on a Macbook Pro M3 laptop, timed using the `process_time` function from the Python `time` module. We report mean and standard deviations measured from 5 different runs, using UMAP embeddings of the MNIST dataset.

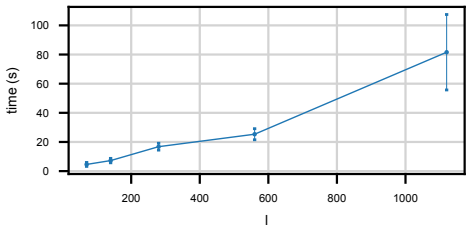
B.11.1 Locality Placement

Locality placement first calculates nearest neighbors for embeddings (size E, dimensionality D_L), runs DIRECT k-means with $k = \ell$, where ℓ is the number of localities, multiple times, evaluating centrality. In $\mathcal{O}$-notation, the `scikit-learn` nearest neighborhood algorithm (KD-tree) used is $\mathcal{O}(D_L\ E\ \log(E))$. The DIRECT method will depend on the distribution of the embeddings in the latent space, but nearest neighborhood calculations for ℓ localities are $\mathcal{O}(\ell\ D_L\ n\ \log(E))$, and the time and centrality calculations $\mathcal{O}(\ell\ n)$, both parallelized over ℓ.

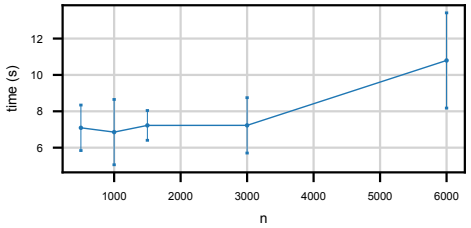
For runtime experiments, we randomly subsample half of the MNIST datasets ($E = 35000$), and set $n = 3000$, $\ell = 140$, and $D_L = 2$ as our default hyperparameter values. We then vary all of the hyperparameters one by one, while fixing the rest to their default values, to estimate their independent effects on runtime. We report results in Fig. S48. As expected from the $\mathcal{O}$-notation analysis, each parameter shows approximately linear dependence with time.



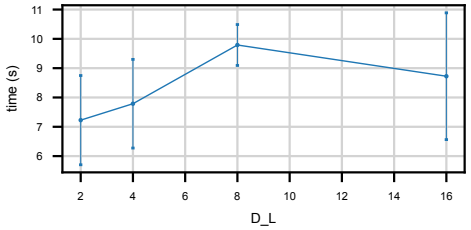
(a) Runtimes when varying the number of samples, E.



(b) Runtimes when varying the number of localities, ℓ.



(c) Runtimes when varying the size of neighborhoods, n.



(d) Runtimes when varying latent dimensionality, D_L.

Figure S48: Runtime mean and standard deviation values for locality placement based on 5 runs.

B.11.2 Locality Representation

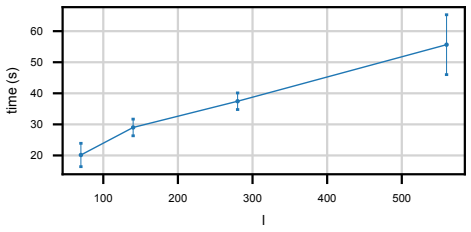
Locality representation calculates pairwise Spearman correlations for D features, for ℓ localities of size n. Ranking the D features for n samples of each of the ℓ localities corresponds to sorting of complexity $\mathcal{O}(\ell\ D\ n\ \log\ n)$, and calculating the Pearson correlations of the ranked data is then $\mathcal{O}(n\ D^2\ l)$. Both are parallelized over ℓ and D, making the calculations scalable. These calculations are standard and we omit runtime experiments.

B.11.3 Module Extraction (AMF)

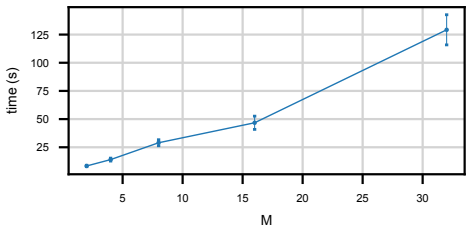
Module extraction is based on our AMF algorithm, which uses stochastic gradient optimization (ADAM) to find M factors in a locality dataset of size $\ell \times \binom{D}{2}$. Therefore, we omit $\mathcal{O}$-notation

analysis. However, parameter-wise, we can expect a linear dependency on the number of modules M, and a quadratic dependency on the number of features D.

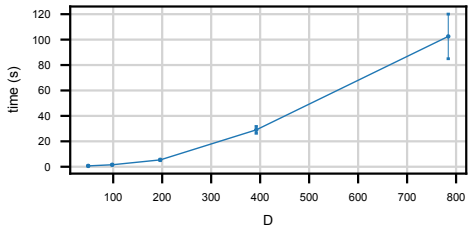
We use default values from the runtime experiments for locality placement. As for the AMF parameters, we use $D = 392$ (50% of the original), $\ell = 140$, $M = 10$ as defaults, and once again vary one by one to estimate independent effects. We report results in Fig. S49. We observe approximately quadratic dependence of runtime on the number of features D, linear with the number of modules M, and sublinear with the number of localities ℓ.



(a) Runtimes when varying the number of localities, ℓ.



(b) Runtimes when varying the number of modules, M.



(c) Runtimes when varying the number of features, D.

Figure S49: Runtime mean and standard deviation values for module extraction based on 5 runs.