# Interactive Hierarchical Label Discovery

by

Meekail Zain

(Under the Direction of Shannon Quinn)

Abstract

Modern machine learning (ML) pipelines, especially deep learning (DL) pipelines, tend to be constrained by the lack of labeled data, whereas raw unlabeled data is relatively abundant. The process of labeling data requires experts to leverage domain knowledge to assign potentially-arbitrary labels to samples. This process is inaccessible to many due to the need of finding a domain expert, as well as overcoming the financial costs of employing such an expert. Furthermore, there is room for error due to accidental mislabeling by the expert. In emerging problems, there may not even be a fundamental set of labels agreed upon by domain experts. Furthermore, labels may be encoded within a hierarchical label schema at different levels of fidelity, leading to sentimental ambiguity. For example, while all shirts are tops, not all tops are shirts. Thus the space of clothing may include labels such as "shirts" and "tops", but whether one is more appropriate than the other is a problem-dependent answer. Sometimes, greater specificity can lead to more complex and confounding models with lower efficacy, whereas being too general may lead to coarser models which do not encode sufficient complexity to model data patterns.

We develop a novel workflow and pipeline to mitigate these problems, built around HDBSCAN which is the current SOTA unsupervised hierarchical clustering machine learning algorithm. We start by modifying HDBSCAN into Path-Constrained HDBSCAN (PCH), a semi-supervised algorithm to allow for expert-sentiment driven hierarchical clustering, which serves to quickly create an initial label schema based on the experts' semantics, amplifying and encoding their personal domain knowledge. We also provide a novel sampling method built specifically for PCH that allows for useful expert queries. We then train a deep representation network designed to produce a rich representation space while also learning representative samples from the data. We then

define a workflow for introspecting the learned samples to gain insights which generalize back to the dataset as a whole.

INDEX WORDS: Machine Learning, Deep Learning, Semi-Supervised Learning, Hierarchical Clustering, Clustering

# Interactive Hierarchical Label Discovery

by

Meekail Zain

B.S., University of Georgia, 2020

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the
Degree.

Doctor of Philosohpy

Athens, Georgia

2025

# Interactive Hierarchical Label Discovery

by

Meekail Zain

| | |
|---|---|
| Major Professor: | Shannon Quinn |
| Committee: | Shannon Quinn<br>Ray Bai<br>Suchendra Bhandarkar |

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
May 2025

# Dedication

I dedicate this dissertation to my mother, whose virtues are outweighed only by her sacrifices – never have I seen someone so completely dedicated to the well-being of others that they have found themselves through it. Everything that I am, is because of you. These achievements are not mine – they are your achievements, and I have the honor of delivering them.

I dedicate this dissertation to my wife, whose self-discovery and growth have set the standards for my own. I have seen you struggle through so much, yet you continue to stand atop your victories and smile with the audacity and bravery to be happy. While I savor the warmth of your companionship, I hope to learn to smile in the same way.

I dedicate this dissertation to every person who has stepped onto the stage of my life, if but even for a moment. As the grains of sand shape a beach, you have shaped me, and I wish I could offer more than these few lines to convey that.

# Acknowledgments

I would first like to acknowledge my wife, Allyson Zain, for stepping up in such immense ways to make the completion of this dissertation as smooth as possible. Thank you for being my teammate and partner through life. Thank you to Quinn Wyner, who after so much proofreading may know this dissertation better than I do. You have been such a vital friend through all these years, and have made an isolating process far less lonely. I want to thank Shannon Quinn for having taken a chance on a somewhat-arrogant freshman a few years ago, and allowing him the freedom to grow and expand in your lab.

To my therapist, and my psychiatrist, and to the mental health workers of this country: thank you. I was in a position where my own mind felt alien to me, and through medication and exhaustive efforts, I was able to learn how to reclaim it, and appreciate it anew. Thank you for helping me get traction on my life again.

Finally, thank you to my mother, who gave me support even when I refused it, and whose pride empowered me to chase what felt impossible, time and time again.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

The conventional process for completing an unlabeled or partially labeled dataset is to employ an expert to create a set of meaningful labels, and partition the dataset according to those labels by assigning each sample an appropriate label. Here, we focus on the prior task of *creating* the set of labels to apply to the data, and term it *label schema discovery*. In some cases, the labels may be obvious, consistent, and well-defined. However, in many difficult tasks, there is not such a clean, canonical choice of labels. Take for example the CIFAR-10 dataset which delineates the following labels: "airplane", "automobile", "truck". Although these three labels can be reasonably distinguished, their utility will depend on the problem context. For instance, if the goal were to distinguish living creatures from animate objects, it may be more appropriate to consider them as a singular class of objects generalized by the label of "vehicle". Conversely, there may be contexts where greater fidelity is appropriate, such as in self-driving car algorithms, and "automobiles" may further be distinguished based on e.g. appearance, form factor, number of wheels, etc. This problem of variable semantic fidelity is especially prevalent in contexts where various experts have opposing views on the subject matter, and thus may prefer different labeling schemata. Thus when referring to a *label schema* we consider not only a final set of labels, but also inter-label relations.

Label schema discovery is generally a qualitative exploratory task undertaken by domain experts, often aided by machine learning tools. For example, an expert may rely on unsupervised clustering tools to estimate the number of distinct clusters within the data, then refine the clusters based on domain-specific knowledge. Unfortunately, due to the complexity of real-world data, these clusters are rarely ideal. Labels often can have *overlap* due to semantic

ambiguity. For example, in sentiment analysis of speech one may distinguish between the sentiments of "angry" and "sad" which linguistically are considered separate sentiments, yet realistically anger and sadness are often intertwined and hence inappropriate for a simple label schema expecting a complete partition.

Ultimately, the most comprehensive process of label schema discovery is to hire a domain expert to manually explore the entire dataset and determine distinct clusters / labels based on domain standards. However, domain experts' time is often limited and therefore valuable, ergo this process can be extremely expensive. Further, the process's immense scope and repetitive nature introduces great potential for human error and arbitrary fidelity in the generated labels due to labels being established *earlier* in the process of exploration potentially biasing the choice of labels *later* in the process for the sake of consistency. Methods that are able to either dynamically evaluate the entire label schema or determine a schema only after considering all the data are resistant to such an issue.

While there exist semi-supervised tools that bridge the gap between manual and fully-unsupervised exploration, they are plagued with several inadequacies. First and foremost, many recent developments in semi-supervised learning focus on large deep-learning systems and frameworks with billions of learnable parameters whose computational costs pose a significant barrier to entry, acting in conjunction with the cost of expert labor. Second, semi-supervised tools can require rather specific feedback formats, such as comparison triplets in the case of metric-learning style semi-supervised techniques, which can raise an additional barrier to entry. For example, asking for metric-learning style comparison triplets may be difficult due to requiring the domain expert to not only distinguish similarity/dissimilarity, but also to impose a consistent partial ordering on the similarity, which is difficult to do a priori. Third, classical semi-supervised machine learning methods such as COP-Kmeans often require the number of assumed clusters as a hyperparameter, making them more appropriate for fine-tuning based on a previously determined estimate of the number of clusters than for initial exploration of a dataset.

To address these shortcomings, we develop a modular, mixed machine-learning / deep-learning framework which interfaces between an expert and a partially or totally unlabeled dataset in order to facilitate the development of a hierarchical labeling scheme that emphasizes expert sentiment. The unique benefits of this framework are that experts do not need to have their own labeling schema for the data a priori and that the complexity of the label schema may develop iteratively based on simple and intuitive feedback.

The bulk contribution of this research will be compatible with an arbitrary vector space. This means that one can leverage any domain-specific feature-extractor (e.g. ResNet, LSTMs, LLMs, etc.) as a backbone to the framework, while the unique learnable pieces of the framework may be trained separately[1]. Thus the framework as a whole disentangles the feature generation and representation steps, allowing for a truly modular solution, greatly expanding the accessibility and applicability of the framework.

[1] Assuming that the backbone network is frozen, though that need not be the case.

Once a representation is constructed, downstream analysis digests the embedded data along with any partial information provided (e.g. a labeled subset, or known inter-data relations) agnostic of the specific original data modality. This analysis results in a new labeling scheme which associates each data sample with a corresponding label (which may not yet 1-1 align with semantic understandings of the data). An expert is then queried by the framework to offer insights on a subset of the data specially curated to inform the labeling schema. The requested feedback is qualitative pairwise similarity: whether two samples are "similar" or "dissimilar" to the expert. The focused scope of the queries transforms an intractable task, asking an expert to comb through an entire dataset to inform a label schema, into a smaller, more tractable task, asking the expert a shortlist of similarity questions. This minimizes the risk of an expert mistakenly offering an incorrect label as feedback, as well as the time and energy that must be invested by the expert in providing domain-specific guidance.

This makes the framework uniquely suited for novel datasets where there are not yet any universally agreed-upon labeling schemes, such as in exploration of a novel disease. The framework may also be useful in cases where there are different subjective understandings of the same data, wherein different schools of thought may seek to define their own taxonomies and labeling schemes. For example, scholars reflecting upon the literary history of a culture may have different opinions on the similarity, and hence underlying structure, of different historical documents. Under this framework, disagreeing scholars could operate over the same underlying data, while producing a labeling scheme unique to their personal understandings of the data through simple feedback. Thus, it may generate potentially disagreeing label schemes which accurately reflect the disagreeing semantics of the experts which generated them, allowing multiple formal and data-driven taxonomies to be developed and explored without the need for a universal consensus.

## 1.2 Formalizing the Problem

### 1.2.1 Clustering

Generally, the task of establishing a set of distinct labels for a dataset is considered clustering. Clustering is a deeply studied part of ML and DL with a rich history. A fundamental aspect of clustering algorithms is the intrinsic trade-off between expert bias (their semantic understanding of what ought to be clustered together, or be distinct, expressed through partial labels and prior information) and the geometric bias of the data (the clusters implied by the data's geometry, dependent on the specific algorithm). There is no correct way to balance this trade-off, and can be considered an expression of one's uncertainty in the expert semantics available for a problem. When there is low confidence in the available expert feedback (e.g. when unsure whether certain expert feedback is useful, or when facing a lack of expert feedback), one may opt to utilize clustering methods which rely more strongly on the geometric bias of the data (e.g. DBSCAN and other fully unsupervised methods). Conversely, favoring the available expert feedback may prompt the use of semi-supervised methods which may incorporate the feedback to influence the geometric insights extracted from the data to varying degrees.

There are a plethora of methods across the spectrum of expert semantic bias vs geometric bias, however almost all of these methods are only capable of producing a *flat clustering* where each label is distinct and totally disjoint from the others.

**Definition.** Given a finite dataset $\mathcal{X}$, a flat clustering on $\mathcal{X}$ is a complete partition of $\mathcal{X}$ written $\mathcal{C} = \{C_1, \ldots, C_k\}$ where $C_i \subseteq \mathcal{X}$ are thus called *clusters*. By definition, we have that $\bigcup_i C_i = \mathcal{X}$ and that $C_i \cap C_j = \varnothing \iff i \neq j$.

We refer to each cluster as being indexed by a separate label, drawing a correspondence between the notions of "labels" and "clusters", often using them interchangeably. While a single sample must have a clear cluster membership under a flat clustering, it may also have various *cluster membership* quantities with respect to the other available clusters. These membership quantities may be heuristic values, intermediate values of the clustering algorithm, or genuine statistical probabilities. A clustering that provides such scores rather than a single index for each sample are referred to as a *soft* clustering, as opposed to a *hard* clustering as described above.

**Definition.** A soft clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a flat clustering paired with an assignment function $f : \mathcal{X} \to [0, 1]^k$ such that $\forall x \in \mathcal{X}\ \sum_i f(x)_i = 1$ and $\operatorname{argmax}_j f_j(x_i) = k \implies x_i \in C_k$.

Note that some algorithms satisfy the partition requirement of a flat clustering by including a "noise" cluster meant to contain any points that the algorithm cannot confidently cluster, or that the algorithm otherwise rejects, whereas other algorithms (such as k-means) guarantee a complete partition without the concept of a noise category.

### 1.2.2 Hierarchical Clustering

This format is commonly used in traditional single-annotation classification problems. This comes with many potential downsides, including the inability to distinguish the relationship between labels which may exist in an ideal semantic analysis. Taking into account the CIFAR-10 example posed earlier, this inability corresponds to the fact that "airplanes", "automobiles", and "trucks" may all be considered "vehicles" in general, while a flat clustering includes them as separate disjoint clusters.

The assumption of strictly disjoint clusters greatly limits the expressiveness of the output clustering. Each point must belong exclusively to a single cluster precluding the clustering from representing multiple direct relationships between points. In other words, there exists a single relation between points in the dataset, offering a sort of "one-dimensional" view regarding their relationships. This problem is alleviated by *hierarchical clustering* methods such as the family of agglomerative clustering methods, and HDBSCAN which encode not only a flat clustering (and hence the corresponding labels) but also the hierarchical relationship between labels, often encoded as a dendrogram (see figure 1.1). More formally, we can define the hierarchy of a given set of labels by representing the labels as a partially-ordered set (poset).

**Definition.** Given a dataset $\mathcal{X}$, a hierarchical clustering on $\mathcal{X}$ is a finite set of clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ with an additional partial ordering where $C_i \leq C_j \iff C_i \subseteq C_j$ that satisfies the following properties:

1. If $C_i \cap C_j \neq \varnothing$, then $(C_i \subseteq C_j) \vee (C_j \subseteq C_i)$.
2. There exists a subset of pairwise disjoint clusters $\{C_{i_1}, \ldots, C_{i_l}\} \subseteq \mathcal{C}$ that covers the dataset. That is, $C_{i_a} \cap C_{i_b} = \varnothing \iff a \neq b$, and $\bigcup_j C_{i_j} = \mathcal{X}$.

The final requirement ensures that we can extract a valid flat clustering from a more general hierarchical clustering.

Note that every flat clustering trivially satisfies the definition of a hierarchical clustering. Similarly, every hierarchical clustering can be consistently extended to include a trivial flat clustering composed of data singletons. Many

hierarchical clustering algorithms heavily leverage these two facts and rely on them to generate either agglomerative ("bottom-up") or divisive ("top-down") methods.

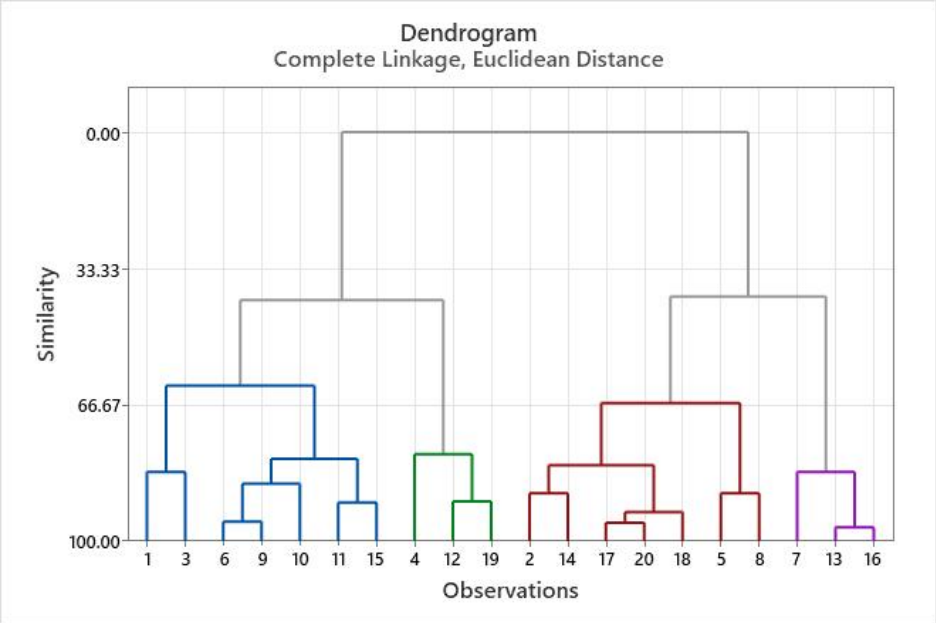


Figure 1.1: An example of a dendrogram encoding the hierarchical relationships between labels. In this example, the labels correspond to the singleton clusters comprised of individual observations. Notice how highly similar clusters such as observations $6, 9$ are merged in the dendrogram into a "virtual" cluster, represented as a horizontal line in the hierarchy. This virtual cluster is then further merged until eventually all clusters are merged, producing a final hierarchical clustering. A single flat clustering can be extracted at an arbitrary level of fidelity by "cutting across" the dendrogram at a fixed similarity score – i.e. merging up to a threshold similarity. The colored groupings represent such a flat clustering at similarity $= 50$.

### 1.2.3 Label Schema Discovery

The objective of "label schema discovery", given an unlabeled or partially labeled dataset, is to then develop a label schema such that the data and their labels correspond to consistent semantic targets as decided by a domain expert. This is slightly different from simply conventionally labeling a dataset, as the expert need not know ahead of time what labels are appropriate for the data. This is a significant advantage when conducting exploratory analysis on novel data that is not yet well studied. Examples of exploring such novel data include discovering new pathologies induced by a genetic knockout, evaluating textual sentiment

across a corpus of historical documents, and even image representation learning on web-scraped data that is unprocessed and highly varied.

We start by constructing a statistical model to represent the available label information on the dataset. This is constructed from the structure of the learned embedding in combination with any explicitly provided feedback, such as initial labels. We must then find a way to effectively propagate the information contained in the labeling onto the unlabeled data. The main complication of this task is that, under our problem formulation, the current label schema may not be surjective onto the label space – that is to say, **we may not yet have all the categories that a "ground truth" labeling would use**.

This is generally an insurmountable barrier for conventional algorithms, where prior-knowledge dictates, to a large degree, the structure of the labeling, if not the space of labels themselves. The primary novelty of this module is that is mitigates that requirement, allowing for implicit label schema development.

A labeling schema $(\tilde{L}, <)$ is defined as a collection of unique class labels $\tilde{L}$ alongside a partial ordering between labels ($<$), indicating an "inheritance" of sorts. If labels $x, y$ satisfy $x \leq y$, then all points labeled $x$ would also be correctly considered instances of $y$, whereas the converse is true only when they are exactly the same label. We may equivalently represent the labeling schema as a poset which encodes a hierarchical labeling system, similarly to how we defined a hierarchical clustering. In fact, both structures are equivalent in the sense that a hierarchical clustering induces a labeling schema by mapping $(\{C_1, \ldots, C_j\}, <) \rightarrow (\{i, \ldots, j\}, <)$ where $x \in C_i$ implies that its corresponding label $y = i$, and that $C_i < C_j \implies i < j$[2].

[2] A reminder that this is the partial ordering of labels named after integers, not of integers themselves.

### 1.2.4 Manifold Hypothesis

Although the information encoded in the labeled data subset is explicit (by way of the labels themselves), a viable framework should ideally utilize the more implicit information encoded in the distribution of *unlabeled* points as well. The information encoded in the distribution of unlabeled points is largely that of the geometric structure of the underlying data distribution. Specifically, we leverage the popular manifold hypothesis to codify this information and analyze it. The manifold hypothesis claims that all data lie in some lower-dimensional Riemannian manifold (a manifold with a well-defined Riemannian metric) embedded within the ambient data space, and that the local distribution of points in an arbitrary neighborhood encodes the local Riemannian metric [22].

We define a manifold $\mathcal{M}$ as a special subset of the ambient space $\mathbb{R}^n$ such that $\mathcal{M}$ is locally linear, enjoying several convenient properties. We focus on Riemannian manifolds, which come equipped with a Riemannian metric $g_p$ :

$\mathrm{T_p(M)} \times \mathrm{T_p(M)} \to \mathbb{R}$ which sends two tangent vectors at a given point $p$ to a scalar value. The Riemannian metric induces a localized norm $\|\cdot\|_p : T_pM \to \mathbb{R}$ defined by $\|v\|_p = \sqrt{g_p(v,v)}$. Note that the Riemannian metric exists for each point $p$ and thus induced vector norm need not be "constant" (with respect to e.g. translation invariance) globally.

Functionally, this translates to two assumptions:

1. The data lie on some subset that has volume zero in the ambient space, but is well-behaved and can be locally approximated linearly.

2. We can define locally-consistent senses of distance on the manifold by the data sampled from it at arbitrary points.

In particular, data is generally understood to be "noisy", represented under the manifold hypothesis as a true data point on the underlying manifold offset by some amount of random noise. This noise is the primary confounding factor of analysis under the manifold hypothesis, since it forces the data to occupy non-zero measure in the ambient space, meaning it is impossible to fit an accurate underlying manifold to the data without significant overfitting (though the recent neural scaling laws have challenged this assumption [40]).

A naïve interpretation of the manifold hypothesis would suggest that local geometry is sufficient to represent the manifold as a whole, however great care must be taken to ensure that global geometry is not sacrificed to better optimize for local geometry, as is documented and observed in methods such as t-SNE which notoriously preserve *local* geometry at the expense of *global* geometry. Indeed a balance of both must be obtained to maximize the accuracy of the data representation. Some contemporary methods such as Uniform Manifold Approximation & Projection (UMAP) are better suited for such tasks.

The manifold hypothesis and its implications can be used to guide the development of methods such as how to use the spatial distribution of unlabeled points to aid in the extension of information encoded in labeled points to more of the data. As an analogy, if a label were to describe the "color" of a point, a naïve way to propagate the information to nearby unlabeled points would be to "pour" some paint of that color starting from the labeled point, allowing it to spread radially outward with respect to the ambient space. However, the manifold underlying a dataset is often nonlinear, such as a curved piece of paper embedded in 3D space. Leveraging the manifold hypothesis, it would be more appropriate to allow the paint to spread radially with respect to the *local geometry*, which can be approximated by the local distribution of points, i.e. distribute the paint as a disk on the paper rather than a ball in the ambient space. Thusly applying the manifold hypothesis in propagating label information over

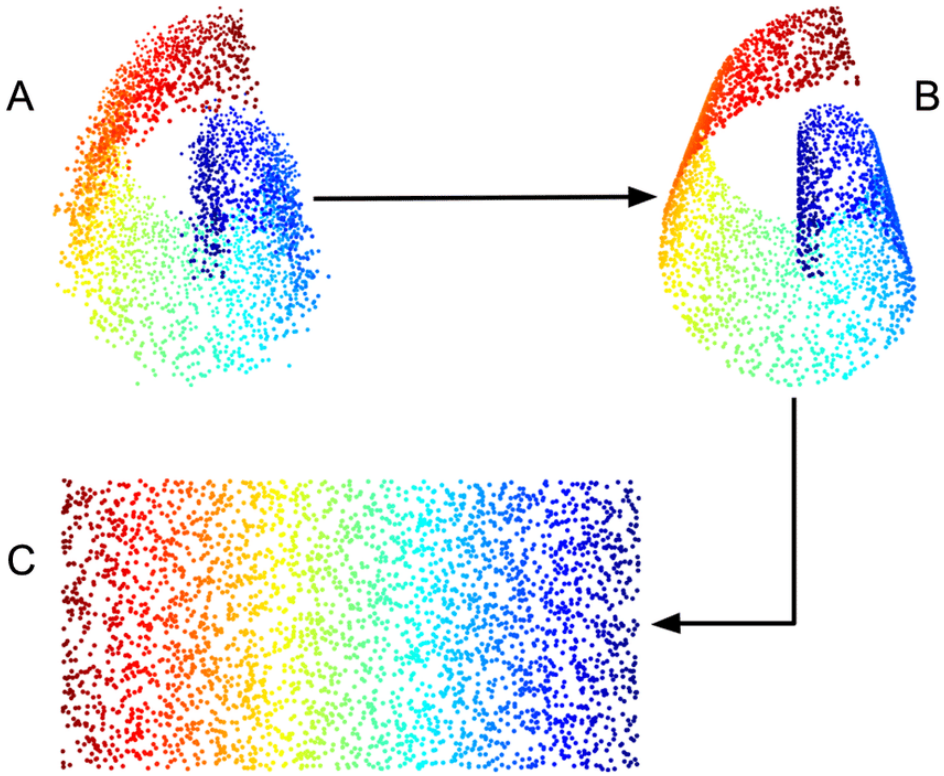


Figure 1.2: A visualization of locality on a data manifold versus its ambient embedding space from [74]. A) shows an embedded dataset in $\mathbb{R}^3$. B) shows the projection of the data onto their underlying manifold as embedded in $\mathbb{R}^3$. C) shows the data manifold itself as an isomorphism to a subset of $\mathbb{R}^2$. Note how spatial locality in the ambient space does not directly correspond to data similarity with respect to the underlying manifold – points on opposite ends of the manifold may be relatively close in the ambient space, yet be semantically different. That difference is best represented with respect to the manifold geometry.

a dataset allows for greater consideration of local spatial information that can provide new insights in exploring the data that a naïve global approach may not.

### 1.2.5 Semi-Supervised Feedback

The format and underlying nature of the solicited feedback greatly affects the efficacy of the framework as a whole. In particular, many conventional algorithms are inadequate for label schema discovery exactly because their feedback requires a fixed label schema a priori, without the ability to extend it as needed. This means that an expert must know ahead of time exactly what groups they

are looking for, and may only learn the distribution of those groups in the data space. Our proposed framework is free from such a limitation, and accomplishes that by relying on simple pairwise similarity constraints. Specifically, we accept pairwise constraints in the form of "Must-Link" constraints (MLC), and "Cannot-Link" constraints (CLC). Standard feedback consisting of explicit labels for specific data can be easily translated to the MLC/CLC format by comparing whether two samples belong to the same cluster or different clusters, respectively. By using this pairwise similarity information, experts can provide feedback to indicate semantically meaningful relationships in the data without having an explicit label-schema a priori.

# Chapter 2

# Path-Constrained HDBSCAN

## 2.1 Introduction

We introduce Path-Constrained HDBSCAN (PCH), a novel, fast, semi-supervised hierarchical density-based clustering algorithm built on top of the state of the art (SOTA) HDBSCAN algorithm [54]. PCH is a lightweight machine learning algorithm which can be utilized even in compute-constrained environments and scales well to large datasets, both in the number of samples and the data dimensionality. PCH serves as a self-contained exploratory algorithm that delivers a robust hierarchical label schema taking into account arbitrary pairwise MLC/CLC.

Whereas HDBSCAN relies entirely on the geometric information provided by the unlabeled dataset to build a global hierarchy based on local mutual reachability distances (an approximate estimate of the local metric tensor), PCH extends this process by further allowing for minimally invasive "virtual" mutations to the data space that encourage not only final clusters which respect the provided constraints, but an *entire cluster hierarchy* that reflects the underlying constraints. This difference distinguishes it from the original suggested implementation of a semi-supervised HDBSCAN, which suffers from being constrained by the original cluster hierarchy of HDBSCAN run on the unlabeled data, limiting the influence of expert feedback. While the original implementation's prioritization of spatial information may be beneficial in cases where the geometric distribution of data corresponds well to the ideal semantic interpretation of the data, as limiting the influence of expert feedback can be seen as limiting the introduction of human error to the system, this is almost never truly the case in practice. Embedding techniques rarely produce such

ideal representations of the data, and consequently this invariance proves to be a liability, whereas the robustness of PCH to poorer spatial representations proves to be an asset.

## 2.2 Related Works

Semi-supervised clustering is a subset of the more general task of clustering where, in addition to raw unlabeled data, we have some form of prior information regarding the relationship between data and labels or between data points themselves. Semi-supervised clustering tasks focus on using this prior information to improve the clustering process, in particular to align the produced clusters with the semantic perspectives encoded in that prior information[93]. For example, given a partially-labeled dataset wherein most of the data is tagged with their corresponding ground-truth labels, extending those labels to the remaining few unlabeled observations would be a semi-supervised clustering task. Another such task would be constructing a clustering over an unlabeled dataset given only sparse, instance-level information regarding the pairwise similarity or dissimilarity between data points [14, 103]. In general, semi-supervised clustering can be understood as the set of clustering tasks wherein prior information is propagated out to raw unlabeled / unannotated data in a way that is semantically consistent and aligns with the task at hand [69, 14]. However, the quantity, format, and reliability of that prior information can vary significantly between two different semi-supervised clustering tasks.

There are many perspectives one may use to analyze the field of semi-supervised clustering, with unique taxonomies resulting from choices to organize by clustering strategy, type of prior information, and even type of mechanism for prior information enforcement [8]. We consider this from the perspective of what type of prior information is incorporated, since a central constraint to our problem is the *lack* of an a priori label schema. Truong et al. partition semi-supervised clustering based on type of prior knowledge into the categories of labels, pairwise constraints, and several other miscellaneous formats, such as prior membership degree and grouping information [86, 95, 62, 97]. From this perspective, it is clear that label-based methods are untenable, since the problem presumes a lack of well-established label schema. Indeed, pairwise constraints and other membership quantities are the only viable options, since pairwise constraints can be as simple as dictating whether a given pair of points is part of an MLC or CLC. These constraints can be generated directly given a set of partial labels, however the converse is not necessarily true. Although a trivial label schema can be generated by the distinct groups of MLCs, it is not necessarily ex-

haustive; there may be entire classes of observations completely unrepresented in the pairwise constraints. Moreover, if there are multiple disjoint groups of MLCs belonging to the same ground-truth label, then naïvely assigning unique labels to each MLC group would fragment the ground-truth class into different clusters, even though the absence of an MLC linking the two groups is not itself an indication of a semantically meaningful distinction between those groups.

Pairwise constraints have been used in a plethora of domains. They're commonly applied in biology, where gene clustering and gene expression data is often parsed based on co-occurrence, and consequently are a natural source of abundant MLCs/CLCs [21, 70, 90]. They are also commonly used in semi-supervised clustering of textual datasets on account of the fact that textual datasets often present significant semantic ambiguity regarding pairwise connectivity of documents, which pairwise constraints can help clarify to ensure consistent clustering [58].

In addition to the prior-information based taxonomy, we also consider the separation of hierarchical and flat clustering techniques. Cai et al. note that "most semi-supervised hierarchical clustering methods are the [variants] of the single link, complete link, and average link method" which are cases of linkage methods, augmented further by Ward's Linkage and Centroid Linkage [80]. The linkage methods are relatively straightforward themselves, but generally do not perform as well as other pairwise-constrained methods such as the constrained k-means (COP-Kmeans) algorithm, and its related family of k-means based algorithms such as PCKmeans, MPCKM, SSKFCM, and SCKMM. These are all generalizations of k-means developed to optimize for pairwise constraints, yet they are ultimately limited by the linear nature of k-means [4, 7, 101, 96]. Although they can be effective in various problem contexts, they are limited in the solution space they cover such that semantic groupings which are not implied by or even violate the spatial distribution of data are out of reach. Due to the mean-focused nature of the algorithms, they are also sensitive to the underlying density and distribution of points, failing in cases of non-convex clusters [4, 101].

Instead of relying on the existing selection of limited semi-supervised hierarchical models, we turn to two potential options. We may either develop an existing semi-supervised flat clustering algorithm to produce a hierarchical clustering, or modify an existing hierarchical clustering algorithm to produce semi-supervised outputs. To that effect, we consider Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) as proposed in [10]. HDBSCAN is a fast, effective, SOTA fully-unsupervised hierarchical clustering algorithm that is an incredibly popular tool used widely across domains.

Its popularity is largely due to an accelerated implementation by McInnes et al. which was then up-streamed into scikit-learn by Zain et. al [54].

While HDBSCAN has remarkable general use-case efficacy and is an invaluable exploratory tool, it lacks the capacity to incorporate prior information into its clustering procedure. Campello et al. of HDBSCAN included a theorized specification for a semi-supervised variant in their original paper [10]. This variant is theoretically limited and has a narrow solution space insufficient for real-world complexity. In particular, while it is not as tightly constrained around linearly-separable clusters as methods such as k-means are, it exhibits strict adherence to its own cluster hierarchy fully determined by the spatial distribution of points, and is inflexible in assigning clusters that would otherwise violate this hierarchy. This is discussed in further detail in later sections.

We use HDBSCAN as the basis for our novel semi-supervised PCH to avoid the assumption that all clusters on the data manifold have points distributed with a relatively-uniform density.

## 2.3 HDBSCAN Review

### 2.3.1 Mutual Reachability

Given a dataset $\mathcal{X}$ in a metric space $\mathcal{R}^n = (\mathbb{R}^n, d)$ and a fixed parameter $k \in \mathbb{Z}_+$ which defines the locality of the algorithm, HDBSCAN begins by defining the notion of a point's "$k$-core distance"

$$d_c(x) = d(x, x_*)$$

where $x_*$ is the $k$-th nearest neighbor of $x$, where $x, x_* \in \mathcal{X}$. Since we keep $k$ fixed, we elide its mention and succinctly refer to $d_c$ as the *core distance*. This core distance serves as a local estimate of the metric tensor $g$ for the underlying data manifold at a given point $x \in \mathcal{X}$, denoted $g(x)$.

This estimate is based on the assumption that the neighborhood $B(x, d_c(x))$ centered at $x$ will, for a small-enough choice of $k$, be local enough that $g$ is approximately constant, relying on the locally-euclidean nature of the Riemannian manifold hypothesis. Note that this locality and consistency of the local metric tensor is better guaranteed for small choices of $k$ corresponding to taking the limit of the metric tensor as the neighborhood radius approaches $\|x - x_{k_1}\|$ where $x_{k_1}$ is the nearest-neighbor of $x$. However, because data generally consists of "noisy" off-manifold points rather than an idealized lattice of points positioned precisely on the underlying manifold, small choices of $k$ also become

more prone to reflecting the noise intrinsic to the data set rather than the actual metric tensor corresponding to the underlying manifold.

Conversely, taking larger values of $k$ reduces this noise, at the expense of the metric tensor reflecting an 'average' value across a larger region containing $x$ rather than the instantaneous value at $x$ as desired. In short, small choices of $k$ exhibit lower bias and higher variance, whereas large values exhibit higher bias but lower variance. It is important to consider the fact that any given dataset is a finite collection of samples under the data distribution, and thus the relative local density of points may differ greatly, resulting in different optimal choices of $k$ for datasets of different density (number of samples). In general, higher-density datasets afford higher values of $k$ for an equivalent amount of bias, and conversely a lower $k$ for an equivalent amount of variance.

The core distances are then used to compute pairwise *mutual reachability* distances (MRD). The MRD between points $x, y \in \mathcal{X}$ is defined as

$$d_m(x, y) = \max\{d_c(x), d_c(y), d(x, y)\}$$

which is generally a conservative estimate of the manifold-distance between $x, y$ at local scales, while ambient-space distance dominates at global scales. Note that the MRD is a proper *metric*, making $(\mathcal{X}, d_m)$ a metric space. The use of MRD is a significant factor in how HDBSCAN manages to combine local and global geometric information. Refer to figure 2.1 for a visual representation of core-distances and MRD.

### 2.3.2 Minimum Spanning Tree

To further refine the geometric information contained within the pairwise MRDs, HDBSCAN constructs a minimum spanning tree (MST) denoted $G$ using the MRDs as edge weights, essentially charting the local-connectivity of every point to its neighbors as mediated by their local metric tensors and estimated by MRD. The process is a straightforward MST building process, e.g. utilizing Prim's algorithm[3]. The key property of $G$ is that, by the definition of an MST, removing $m$ edges from $G$ results in exactly $m + 1$ disjoint connected components. This insight seems trivial at first glance, but the ingenuity of HDBSCAN lies in the connection of this fact to two commonly adopted assumptions about well-defined *clusters*: data should be dense *within* clusters, and sparse *between* clusters. Granted that the edges of $G$ encode density, HDBSCAN generates a hierarchy starting with a singular universal data cluster. This cluster is given an initial label, and is the first of many eventual clusters in the hierarchical clustering.

[3] In practice HDBSCAN has both a Prim's algorithm implementation and a Borůvka's algorithm implementation. These algorithms are optimal in different cases depending on dataset size and dimensionality, yet ultimately both produce an MST.

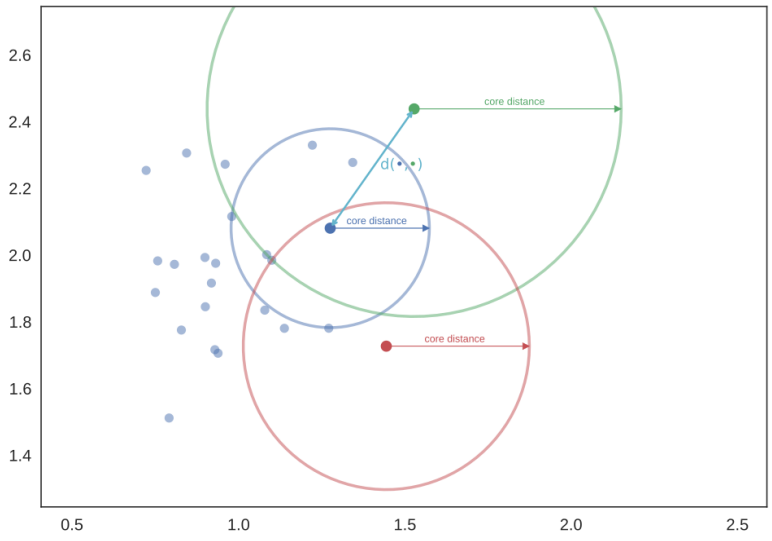


Figure 2.1: A visualization of the relationship between points' relative spatial distributions and their core distances, and consequently their MRD. Note that the MRD between the blue and green points is equal to the core-distance of the green point, since it is larger than the blue point's core-distance, and larger than their ambient spatial distance. This is an example of how points that are closer than the $k$-th nearest neighbor are essentially "pushed" outwards and treated as being at least core-distance away for the MRD calculation, as the blue point was here for the green point.

The MST is then cut by removing the *largest* edge in $G$, bisecting it into two disjoint components. If the smaller of the two components is *sufficiently large*, then it is considered a "new cluster", and both clusters are given new labels. If, the smaller of the spawned clusters is too small, then the disjoint component is considered a collection of "noise", implying that rather than a meaningful cluster being found, noisy off-manifold points were separated off of a larger salient cluster. This means that no new labels are created, and there is no change to the cumulative hierarchy thus far. This process repeats until *all* edges are cut, resulting in an exhaustive hierarchy culled by removing "noisy" splits. [4]

[4] In practice, this is actually computed "backwards" in that the hierarchy is generated *agglomeratively* starting from singleton sets of individual data points and building up based on the *smallest* edges in the MST with non-noise labels only being issued once the clusters are merged up to sufficient sizes. This is functionally identical, but a bit more complex to explain precisely.

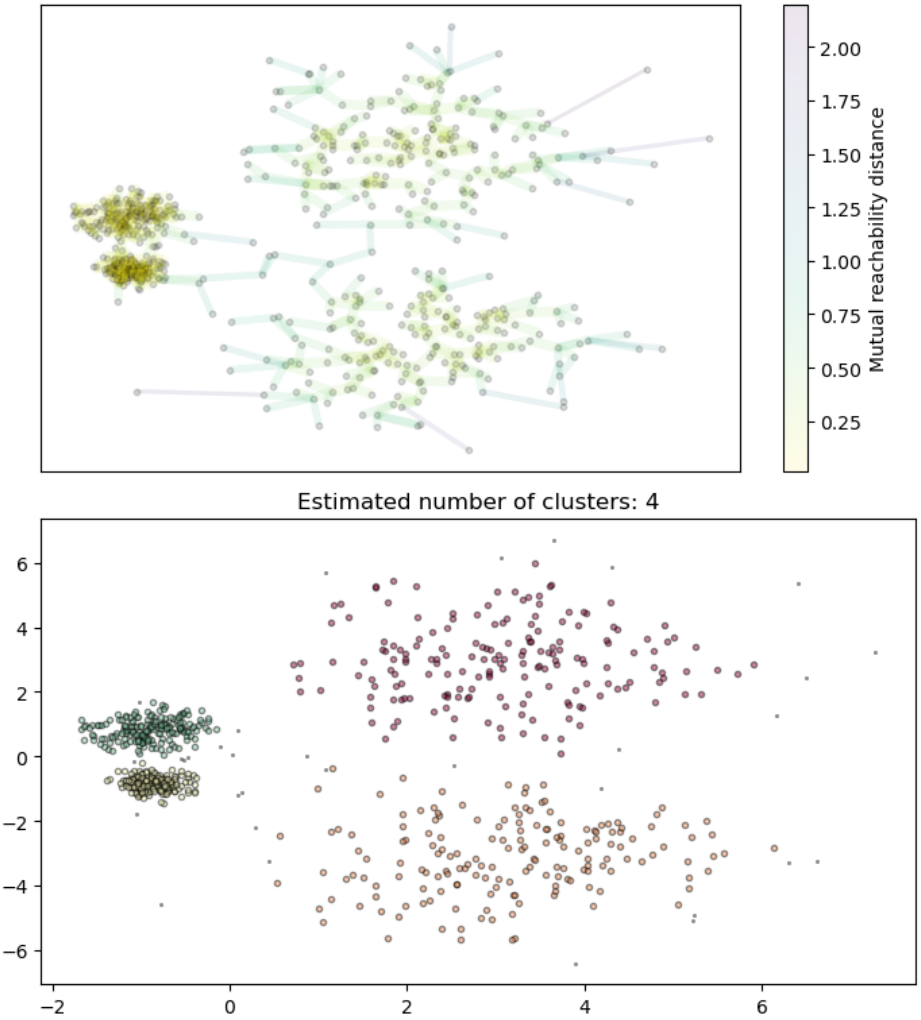


Figure 2.2: A visualization of an MST with edges colored by their relative weights (MRD).

### 2.3.3 Cluster Selection

This hierarchical clustering represents "all possible" clusterings according to the leaf clusters it has picked up on. To select a singular flat clustering from this hierarchy, HDBSCAN relies on the heuristic measure known as the "excess of mass" (EOM) method, which measures the persistence of clusters across the hierarchy, selecting clusters that had lived for longer than their descendant clusters as final nodes in a flat clustering. For details on the EOM algorithm, please refer to [10, 11, 54].

### 2.3.4 Semi-Supervised Cluster Selection

The proposed solution to a semi-supervised HDBSCAN as mentioned in Campello et al. is to modify the cluster selection mechanism to instead optimize for the number of constraints satisfied (hereafter referred to as the "baseline method") [11]. A given flat clustering generated from the hierarchical clustering is evaluated based on the number of constraints that such a clustering satisfies, with the convention that any point labeled as noise automatically satisfies any CLC it participates in, whereas it fails any MLC. Consequently, the baseline method is tightly bound to the hierarchical clustering generated by HDBSCAN, and is unable to override the structure of such a clustering. While HDBSCAN extracts well-structured local geometric information in a globally consistent way, the baseline method lacks the ability to interfere with the construction of its clustering hierarchy based on external feedback. If two clusters satisfy constraints when merged together, their method can *only* merge them through selecting a common ancestor cluster of both of those clusters, which may implicate other clusters that otherwise ought to remain separate. Consequently, the scope of solutions achievable with this method are greatly limited by the spatial locality of clusters, which itself is a downstream product of whatever representation space was used for the data (whether raw or processed) and is sensitive to the quality of the space.

## 2.4 Methodology

To expand the solution space, and develop a robustness to the choice of representation space, we develop a method to intervene at the MST-building stage, rather than the cluster-selection stage. This earlier intervention allows for the construction of unique MSTs which may result in genuinely different hierarchical relationships between perceived clusters, allowing for greater flexibility in final cluster selection.

### 2.4.1 Link Classes

PCH first constructs and maintains a list of disjoint sets representing "transitive must-link classes" which we refer to as "link classes". Each link class is a set of points that not only are directly linked together through an original MLC, but are also linked *transitively* by two or more MLCs. Thus we turn the otherwise direct pairwise relation of must-link constraints into a transitive variant, which forms proper equivalence classes. That is to say, given MLCs $(u, v), (x, y), (v, x)$ for points $u, v, x, y \in \mathcal{X}$ we use the fact that

$(u \sim v) \wedge (v \sim x) \implies u \sim x$ to enumerate that

$$(u \sim v) \wedge (x \sim y) \wedge (v \sim x) \implies (u \sim y) \wedge (v \sim y) \wedge (u \sim x)$$

where we denote a direct or transitive must-link connection between points $p, q$ with $p \sim q$. These equivalence classes derived from a few MLCs lie at the heart of PCH. We optimize not for just the listed MLCs, but the entire set of MLCs implied by the equivalence classes, thus re-incorporating implicit expert feedback without needing to re-query experts directly. This also helps ensure consistency among the explicitly provided MLCs, maximizing constraint satisfaction. We compute these link classes by starting with every data point as its own singleton class. As we iterate through the MLCs, we check to see for each point whether it is already present in a non-singleton set. If both points are present in the same set, then nothing need be done. If instead each point belongs to a distinct non-singleton set, then we simply merge the sets. If only one point belongs to a non-singleton set, we simply add the remaining point to that set. Finally, if no sets contain any of the points, we initialize a new set comprised of the two points. Please refer to Algorithm 2 for implementation details.

### 2.4.2 Path Must-Link Constraints

We start our enforcement of the MLCs by calculating the path between each pair of points in the constraints [5]. This computation is fairly straightforwards, and we utilize a depth-first search across the MST to find the shortest path, recording the edges taken. Once the path for a given constraint is found, we then *truncate* the path, removing initial and terminal edges that directly link two points of the same link class. This ensures that instead of finding a path between two points we instead find a path between the "boundaries" of their *link classes*, which may produce a shorter path. Once a final path is obtained, we calculate the geometric mean of the edge weights on that path, cut the edge with the greatest weight that is not the result of a past MLC or CLC, and introduce a new edge directly between the starting and ending points of the path with weight equal to the previously calculated geometric mean. Note that since we remove a single edge, we create two separate components of the graph, but we immediately "glue" them back together by inserting an edge directly between the start and end points of the path. This essentially "inverts" the path, yet preserves the nature of the graph as a spanning tree[6].

The choice for the weight of the new edge stems from the fact that the MST is not only responsible for the clustering hierarchy, but it also determines the

[5] Recall that since we are operating on an MST, each pair of points has exactly one unique path which is also trivially the *shortest* path between them.

[6] Note that the graph is almost certainly no longer minimal with respect to the original MRD distances, however under special conditions we can guarantee the existence of a coordinate system where it is [16]

outcome of the final flat clustering by affecting the stability of points in the EOM algorithm as determined by their edge weights and local neighborhoods. This means that simply attempting to assign an arbitrarily low value to the new edge weight would essentially "pin" those points in place, greatly strengthening the stability of each of their mutual ancestral clusters, potentially skewing the entire cluster selection outcome. Instead, we assign a weight which does not greatly shift the stability of the points within the path. The least invasive option would be to indeed re-use the weight of the cut edge, but by design we wish to bring these points *closer* together on the MST, and since we targeted the *largest* edge in the path, it would be counterproductive. Thus, we rely on calculating the geometric mean [7] of all edge weights in the path as mentioned before.

[7] While several options were tested, the geometric mean had the best performance and empirically seemed to best preserve the distribution of edge weights. We suspect that this is due to the fact that the geometric mean is more resistant to fluctuations in its inputs, providing a more robust average.

### 2.4.3 Path Cannot-Link Constraints

When implementing CLCs, we have greater liberty in the solution design due to the fact that we need not preserve node/cluster stability since the ultimate goal is to break clusters apart. To this end, we repeat a similar algorithm as for MLCs, except instead of *replacing* the largest edge with a different edge entirely, we simply mutate the largest edge by adjusting its weight. In particular, since the hierarchy is constructed using the ordered edge weights, in order to guarantee separation between points for a CLC, we set the new edge weight to be the old edge weight plus the *greatest* edge weight present in the MST minus the smallest edge weight in the batch of CLCs. This ensures that the first edges to be cut are those that were mutated due to CLC enforcement, guaranteeing the separation of points in all descendant clusters while allowing their original sparsity to dictate their relative ordering.

To prevent a CLC from *erasing* the effects of a prior MLC, as well as avoiding multiple CLCs affecting the same edge, we alter the conditions slightly so as to not just affect the *largest* edge, but rather the largest edge that has not yet been mutated due to an MLC or CLC. For this we must keep a running list of edges which we have mutated. The memory and compute footprint for this is negligible, and we get to ensure consistent and stable MLC and CLC enforcement.

### 2.4.4 PCH and HDBSCAN

An interesting consequence of PCH's early intervention is that it occurs entirely before cluster selection, meaning that it is orthogonal in application to the cluster-selection approach used by [10] in the baseline semi-supervised HDB-

**Algorithm 1** ReadyPrune

**Require:** $(x, y)$, a pair of end-points. $E$ a collection of edges in the MST, with edges in the form $E \ni e = (u, v, w)$ where $u, v$ denote their starting and ending node, while $w$ denotes the edge weight. $S$, a set of link classes.

**Ensure:** Edges in $E$ are marked to be pruned if they are end-points with neighbors closer to the first boundary.

$P \leftarrow \mathrm{DFS}(E, x, y)$
$A \leftarrow \mathrm{LinkClass}(S, x)$
$B \leftarrow \mathrm{LinkClass}(S, y)$
$a \leftarrow x$
$b \leftarrow y$
ToPrune $\leftarrow [F, \ldots, F]$
Initialize $c$, a placeholder for the largest edge.
**if** $A = B$ **then**
    break
**end if**
**for** $e = (u, v, w) \in P$ **do**
    **if** $u \in A \wedge v \in A$ **then**
        ToPrune$[e] = T$
        $a \leftarrow v$
    **else**
        break
    **end if**
**end for**
**for** $e = (u, v, w) \in \mathrm{Reverse}(P)$ **do**
    **if** $u \in B \wedge v \in B$ **then**
        ToPrune$[e] = T$
        $b \leftarrow v$
    **else**
        break
    **end if**
**end for**
    **return** ToPrune

**Algorithm 2** PCH MLC enforcement

**Require:** $(x, y)$, a must-link connection between points $x, y$. $E$ a collection of edges in the MST, with edges in the form $E \ni e = (u, v, w)$ where $u, v$ denote their starting and ending node, while $w$ denotes the edge weight. $S$, a set of link classes.

**Ensure:** $E$ is altered with minimal changes to include an appropriately-weighted direct edge between $x$ and $y$'s link classes through nodes $a, b$ while remaining an MST.

$P \leftarrow \mathrm{DFS}(E, x, y)$
$A \leftarrow \mathrm{LinkClass}(S, x)$
$B \leftarrow \mathrm{LinkClass}(S, y)$
$w' \leftarrow 1$
$w_{max} \leftarrow 0$
ToPrune $\leftarrow$ ReadyPrune$(E, x, y, A, B)$
$c \leftarrow \varnothing$
**if** $A = B$ **then**
    break
**end if**
**for** $e = (u, v, w) \in P$ **do**
    **if** ToPrune$(e)$ **then**
        $\mathrm{Remove}(P, e)$
    **else**
        $w' \leftarrow w' \cdot w$
        **if** $w > w_{max}$ **then**
            $w_{max} \leftarrow w$
            $c \leftarrow e$
        **end if**
    **end if**
**end for**
$w' \leftarrow \sqrt[|P|]{w'}$
$\mathrm{Cut}(E, c)$
$\mathrm{Insert}(E, (a, b, w'))$

**Algorithm 3** PCH CLC enforcement

**Require:** $(x, y)$, a CLC between points $x, y$. $E$, a collection of edges in the MST, with edges in the form $E \ni e = (u, v, w)$ where $u, v$ denote their starting and ending node, while $w$ denotes the edge weight. $S$, a set of link classes. $W$, the largest edge weight present in the MST. Lists of prior MLC/CLC edges, $\mathcal{E}_M, \mathcal{E}_C$ respectively.

**Ensure:** $E$ is altered so that the largest edge on the path between $x$ and $y$'s link classes is updated to have maximal weight, excepting edges in $\mathcal{E}_M \cup \mathcal{E}_C$.

$P \leftarrow \mathrm{DFS}(E, x, y)$
$A \leftarrow \mathrm{LinkClass}(S, x)$
$B \leftarrow \mathrm{LinkClass}(S, y)$
$w_{max} \leftarrow 0$
ToPrune $\leftarrow$ ReadyPrune$(E, x, y, A, B)$
$c \leftarrow \varnothing$
**for** $e = (u, v, w) \in P$ **do**
  **if** ToPrune$(e)$ **then**
    $\mathrm{Remove}(P, e)$
  **else**
    **if** $w > w_{max} \wedge e \notin (\mathcal{E}_M \cup \mathcal{E}_C)$ **then**
      $w_{max} \leftarrow w$
      $w' \leftarrow w + W$
      $c \leftarrow e$
    **end if**
  **end if**
**end for**
**if** $c \neq \varnothing$ **then**
  $\mathrm{UpdateWeight}(c, w')$
  $\mathcal{E}_c \leftarrow \mathcal{E}_c \cup \{c\}$
**end if**

SCAN method. In practice, we have the option of utilizing *both* and indeed find it to be a superior choice.

## 2.5 Metrics

The difficulty of evaluation stems from the fact that most metrics are designed as either *heuristics* regarding cluster geometry – such as the Silhouette score [67], V-Measure, completeness, homogeneity [66], the Calinski and Harabasz score [9] and the Davies-Bouldin score [19] – or as metrics against a ground-truth label set. Consequently, these methods are satisfied by a *unique* label schema determined by the ground-truth labels, and are unable to offer insight on the viability of an arbitrary label schema. In particular, this makes it difficult to evaluate the efficacy of a hierarchical clustering, which at a given extracted flat clustering may not be optimal for a given ground-truth label set, yet may *contain* an ideal clustering as part of its hierarchy. Indeed, no well-studied metrics satisfy this niche, thus we approximate it by focusing on two metrics: *constraint satisfaction* (CS), wherein we calculate the percent of provided constraints which are satisfied in the final labeling, and the Adjusted Rand Index (ARI), which generally is an indicator of the pairwise similarity between two flat clusterings of arbitrary size without any expectation that the clusterings must abide by the same label schema [35].

### 2.5.1 Constraint Satisfaction

Given a flat-clustering $\mathcal{C} = \{C_1, \ldots, C_n\}$ where $C_1$ defines a "noise" cluster, we can define a cluster-membership function $\delta$ as

$$\delta(x, y) = \begin{cases} 1 & \exists C \in \mathcal{C} \setminus \{C_1\} \mid x \in C \wedge y \in C \\ 0 & \nexists C \in \mathcal{C} \setminus \{C_1\} \mid x \in C \wedge y \in C \\ 0 & x \in C_1 \vee y \in C_1 \end{cases}$$

Then with a set of $r$ MLCs $\mathrm{ML} = \{(x_{m_1}, y_{m_1}), \ldots, (x_{m_r}, y_{m_r})\}$ and $s$ CLCs $\mathrm{CL} = \{(x_{c_1}, y_{c_1}), \ldots, (x_{c_s}, y_{c_s})\}$, we can calculate CS as follows:

$$\mathrm{CS}(\mathcal{C}, \mathrm{ML}, \mathrm{CL}) = \frac{1}{|\,\mathrm{ML}\,| + |\,\mathrm{CL}\,|} \left( \sum_{p \in \mathrm{ML}} \delta(p) + \sum_{p \in \mathrm{CL}} (1 - \delta(p)) \right)$$

### 2.5.2 Adjusted Rand Index

Unlike constraint satisfaction, the ARI requires a ground-truth reference clustering $\mathcal{D} = \{D_1, \ldots, D_m\}$, which clusters over the same number of elements as $\mathcal{C}$ does.

$$\sum_{C \in \mathcal{C}} |C| = \sum_{D \in \mathcal{D}} |D| = n$$

No such clustering will be available in an actual use-case of *developing* a novel label schema, however in the case of controlled / labeled datasets, we can use their ground-truth labels directly, or an arbitrary labeling derived as an agglomeration of ground truth labels. We consider a generalized cluster membership function that does not account for noise

$$\delta(x, y; \mathcal{C}) = \begin{cases} 1 & \exists C \in \mathcal{C} \mid x \in C \wedge y \in C \\ 0 & \nexists C \in \mathcal{C} \mid x \in C \wedge y \in C \end{cases}$$

in which case the unadjusted rand index is defined as

$$\mathrm{RI}(\mathcal{C}, \mathcal{D}) = \frac{1}{\binom{n}{2}} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X} \setminus \{x\}} \delta(x, y, \mathcal{C})\delta(x, y, \mathcal{D}) + (1 - \delta(x, y, \mathcal{C}))(1 - \delta(x, y, \mathcal{D}))$$

From here, the ARI is calculated as

$$\mathrm{ARI}(\mathcal{C}, \mathcal{D}) = \frac{\mathrm{RI}(\mathcal{C}, \mathcal{D}) - \mathbb{E}[\mathrm{RI}]}{1 - \mathbb{E}[\mathrm{RI}]}$$

Alternatively, we can refer to the permutation definition which relies on the contingency between clusters $\mathcal{C}, \mathcal{D}$ defining the co-occurrence values $n_{ij} = |C_i \cap D_j|$. We define the row and column cumulants as $a_i = \sum_j n_{ij}$ and $b_j = \sum_i n_{ij}$, allowing us to formulate

$$\mathrm{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

From the permutation formulation, we can also see that if we were to enumerate all $\binom{n}{2}$ possible pairwise constraints for a given dataset and evaluate their enforcement, we would arrive at the ARI. Thus, as the number of pairwise constraints increases, the constraint satisfaction approaches ARI. In this way, one may view ARI as a score of "constraint extrapolation" over the unconstrained portion of the data, as opposed to constraint satisfaction which evaluates only the constrained portion. With that being said, this parallel can only be relied

on at exceedingly high amounts of constraints, since otherwise the ARI will be more indicative of the underlying unlabeled clustering than the semi-supervised component on account of the vast difference in how many pairs of points do, or do not have constraints. For a dataset of size $n$ with $k$ constraints, the number of unconstrained pairs of points will be $\approx \binom{n}{2} - k = O(n^2 - k)$. This decreases if we consider implicit constraints as well (as done in TCS). A naïve approach to estimating the number of unconstrained points would be to assume that every point involved in a constraint must be implicitly involved in a constraint with every *other* point, leading to $O(k^2)$ implicit constraints. Unfortunately, while we can comfortably extrapolate MLCs due to the transitive nature of the constraint, the same is not true for CLCs. Instead, a CLC between two link classes may induce CLCs between each unique pair of points across the two link classes.

If we have $l$ link classes, the number of implicit MLCs can be approximated by considering the average number of constraints per link class is $O(\frac{k}{l})$, and with each constraint adding exactly one point to the link class, except the first which adds two, we have that the average number of points in a link class is $s = O(\frac{k}{l} + 1) = O(\frac{k}{l})$. Noting that every pair of points in a link class form an MLC, hence for $l$ link classes with $s$ points each, we get $O(l\binom{s}{2}) = O(l(\frac{k}{l})^2) = O(k^2 l^{-1})$ MLCs. For $O(k)$ CLCs, we have $O(k(\frac{k}{l})^2) = O(k^3 l^{-1})$ implied CLCs.

Thus the number of unconstrained pairs of points will be $O(n^2 - k^3 l^{-1})$.

The main consequence of this fact is that while constraint satisfaction in its limit is equivalent to ARI, the relationship does not hold at the majority of scales encountered in practice and ARI cannot be understood to "simply" be a generalized version of constraint satisfaction as it will generally reflect more of the performance of the underlying unsupervised component than the semi-supervised interference. To offset this, instead of directly evaluating ARI, we evaluate ARI *gain* as calculated against a reference value of vanilla HDBSCAN run on the same dataset, thereby indicating the improvement to score offered by the semi-supervision, *regardless* of the difficulty of the underlying clustering.

### 2.5.3 Geometric Heuristics

Note that although metrics that provide heuristic scores for "desirable" geometric qualities in clusters are commonly used, they are inappropriate for this context due to the fact that what they evaluate, roughly speaking, is the compatibility of the spatial distribution of points, and their labels and emergent clusters. While this makes sense in many cases and is a valuable metric for most, here it does more harm than good since a primary motivation of our strategy

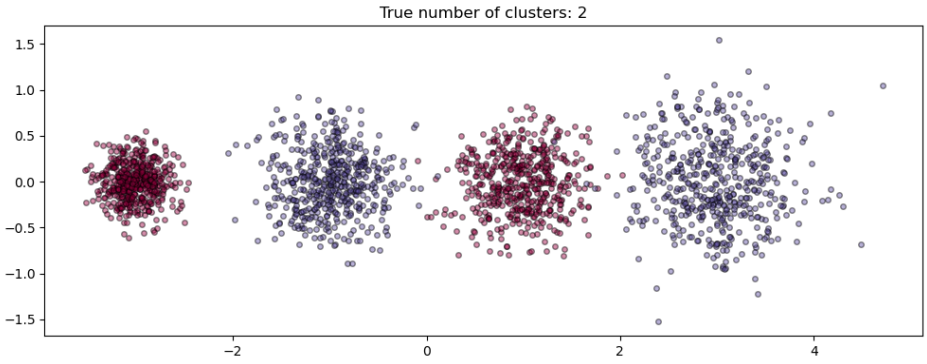


Figure 2.3: An example of an impossible cluster assignment under normal HDBSCAN rules, which becomes achievable under PCH.

is the understanding that the underlying spatial distribution of points *may not be good enough* to provide the clustering required, and thus the mutation of the MST allows us to circumvent such a restraint. Naturally, the geometric indicators of a "good" cluster may not be compatible with the clusters formed from this method, and do not score a clustering's accomplishment of our goal, discovery of clusters consistent with the provided constraints.

## 2.6 Results

### 2.6.1 Synthetic Linear Dataset

We begin by considering a few synthetic cases to highlight the limitations of HDBSCAN's baseline semi-supervised method, as well as the ability of PCH to overcome those restrictions by restructuring the underlying MST. In particular, we construct a simple case of four evenly spaced unit normal distributions with clear separation in a straight line. We set the ground-truth labels to be such that odd unit normals are part of the same class (note that classes need not be 1-1 with spatial clusters), and the even are part of a second class, as seen in figure 2.3. Traditional HDBSCAN, and hence the semi-supervised HDBSCAN baseline algorithm, cannot achieve this clustering because it violates the partial ordering imposed by the standard hierarchy-building algorithm. This can be visualized by looking at the dendrogram graphs for HDBSCAN's clustering hierarchy such as in figure 2.4. HDBSCAN forms the hierarchy in a partial order based on the left-to-right order of the distributions, and is unable to alter the order. Meanwhile, PCH alters the MST directly and hence can engineer a separate clus-

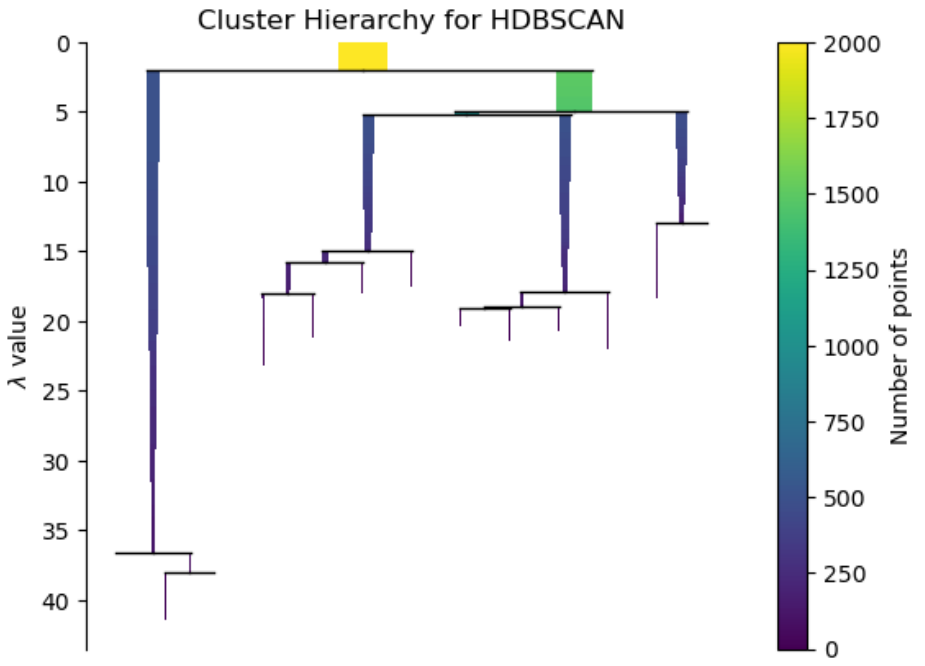


Figure 2.4: The clustering hierarchy dendrogram for HDBSCAN on a linear sequence of clusters with mixed ordering. Each vertical icicle represents a group of points which slowly fade into noise as the density level ($\lambda$, the inverse of MRD) increases. The width and color of the icicle determines their number of points. Each horizontal line represents the split of a cluster into smaller salient groups, starting with the universal cluster at the top.

tering hierarchy which can allow for greater freedom in final cluster selection, as seen in figure 2.5 where the persistent clusters (icicles) are "rotated/pivoted" into different positions with respect to the universal cluster, demonstrating the freedom of arrangement afforded by PCH. Finally, we numerically validate our qualitative findings by considering the ARI of the produced labelings in figure 2.6. Note how the original baseline *suffers* from added constraints, whereas PCH is stable and improves the ARI.

### 2.6.2 Synthetic Antagonistic Dataset

Stepping up the difficulty a bit, we now consider the case of two pairs of Gaussian distributions, where the true class labels are aligned with one Gaussian from each pair, violating expectations of spatial locality as well as challenging partial-ordering as before. The ground truth can be seen in 2.7. HDBSCAN's

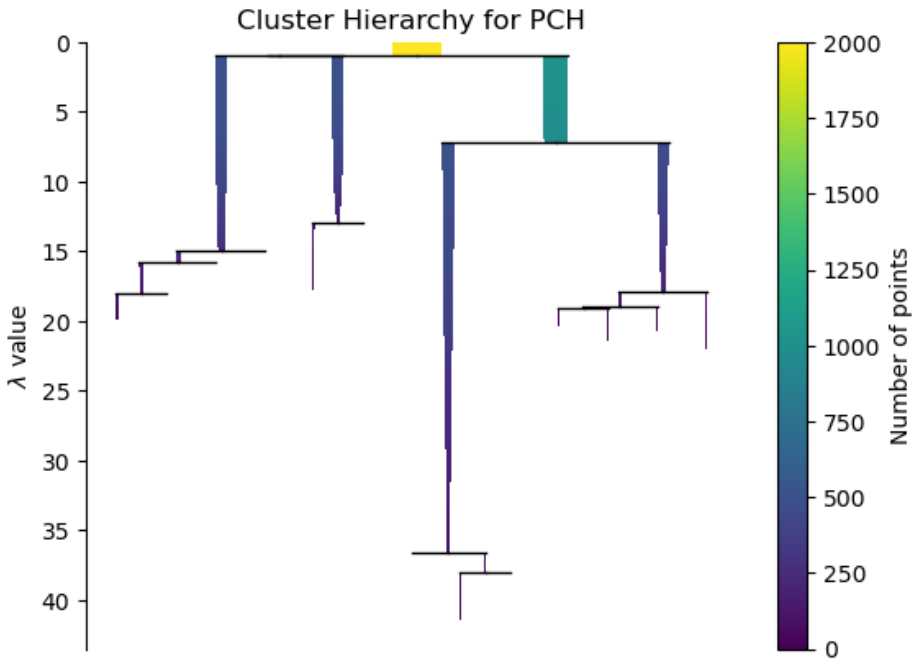


Figure 2.5: The clustering hierarchy dendrogram for PCH on a linear sequence of clusters with mixed ordering.

cluster hierarchy and corresponding cluster assignment can be seen in figures 2.8 and 2.9 respectively. Contrasted with the same components from PCH in figures 2.10 and 2.11, it is clear to see how PCH is able to significantly alter the clustering hierarchy such that a new appropriate flat clustering can be selected by the underlying HDBSCAN algorithm, while preserving the unique approach of HDBSCAN and the insights generated by it. We can see in figure 2.12 that across the board, regardless of the number of constraints, PCH holds a strong advantage over the baseline method which *cannot* produce results any better than *unsupervised* HDBSCAN due to being limited by the shared cluster hierarchy.

### 2.6.3 Wine Dataset

Moving on from synthetic data, we investigate the efficacy of PCH on real data. In particular we begin with the wine dataset [60], a very common dataset in machine learning contexts for both its simplicity, and the difficulty of clustering. This is partly due to the fact that ultimately the labels on the wine dataset are based on highly subjective factors, namely an aggregate "wine quality" whereas

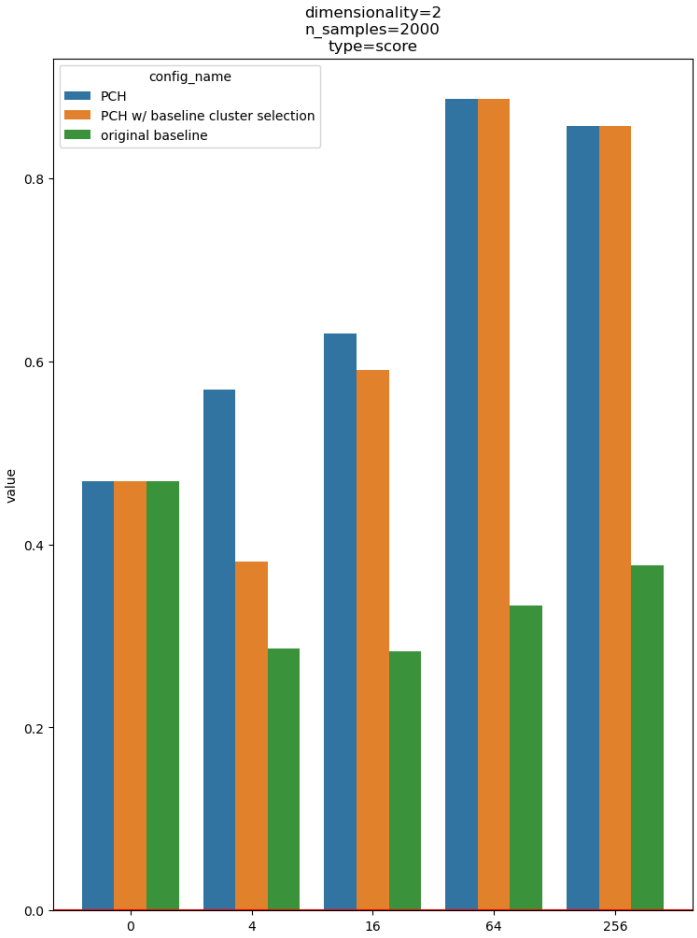


Figure 2.6: The ARI across several methods on a simple linear out-of-order class assignment problem.

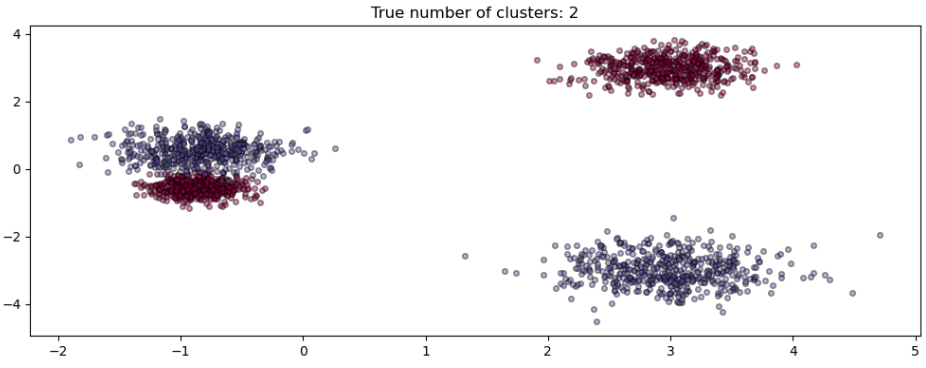


Figure 2.7: A more complex example of an impossible-to-satisfy ground-truth under HDBSCAN which becomes possible under PCH.

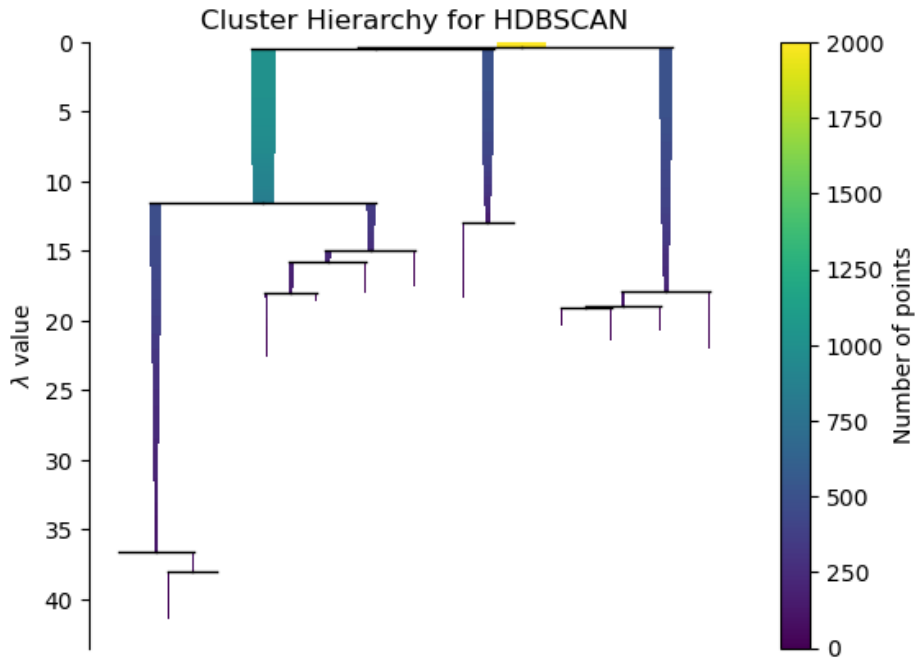


Figure 2.8: The cluster hierarchy of HDBSCAN on an antagonistic dataset.

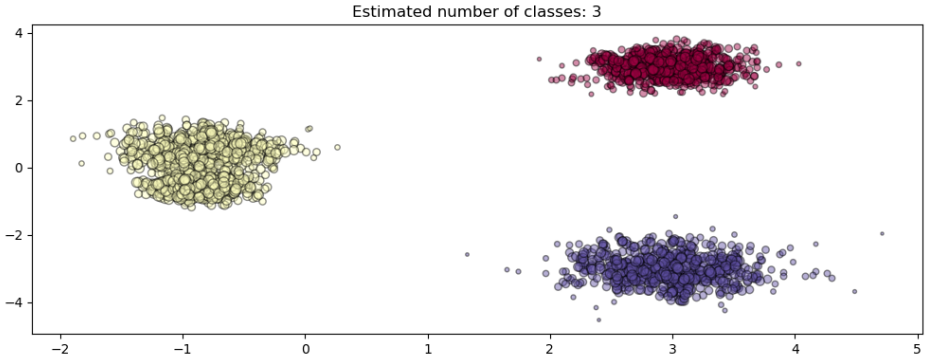


Figure 2.9: The final cluster selection of HDBSCAN on an antagonistic dataset.

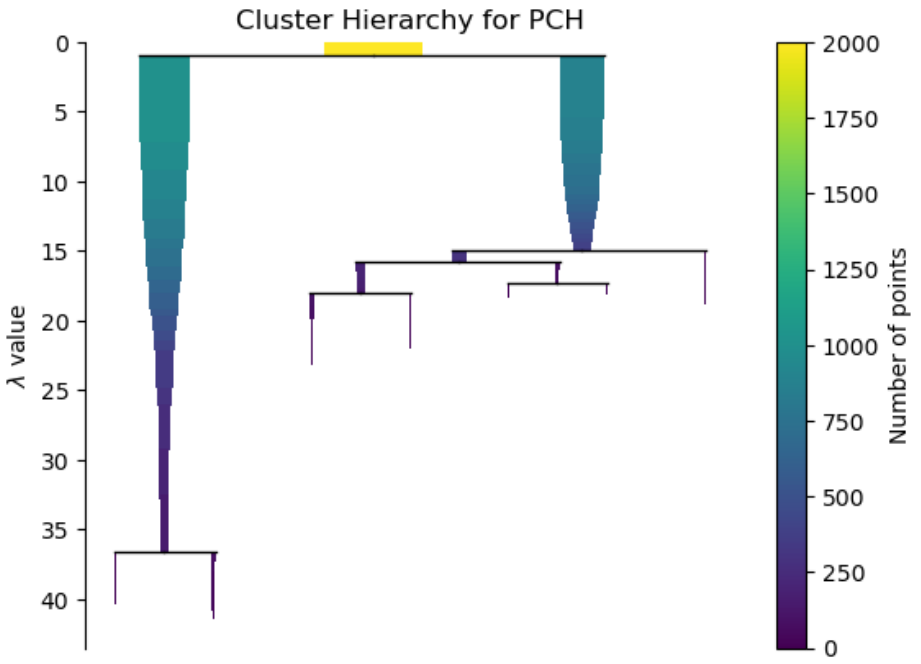


Figure 2.10: The cluster hierarchy of PCH on an antagonistic dataset.

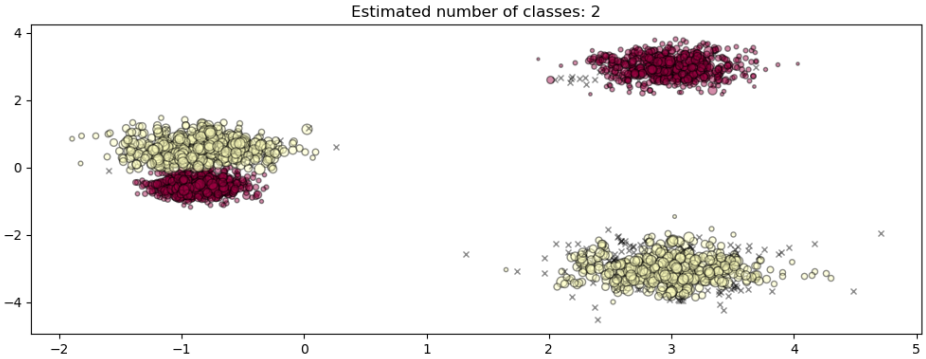


Figure 2.11: The final cluster selection of PCH on an antagonistic dataset.

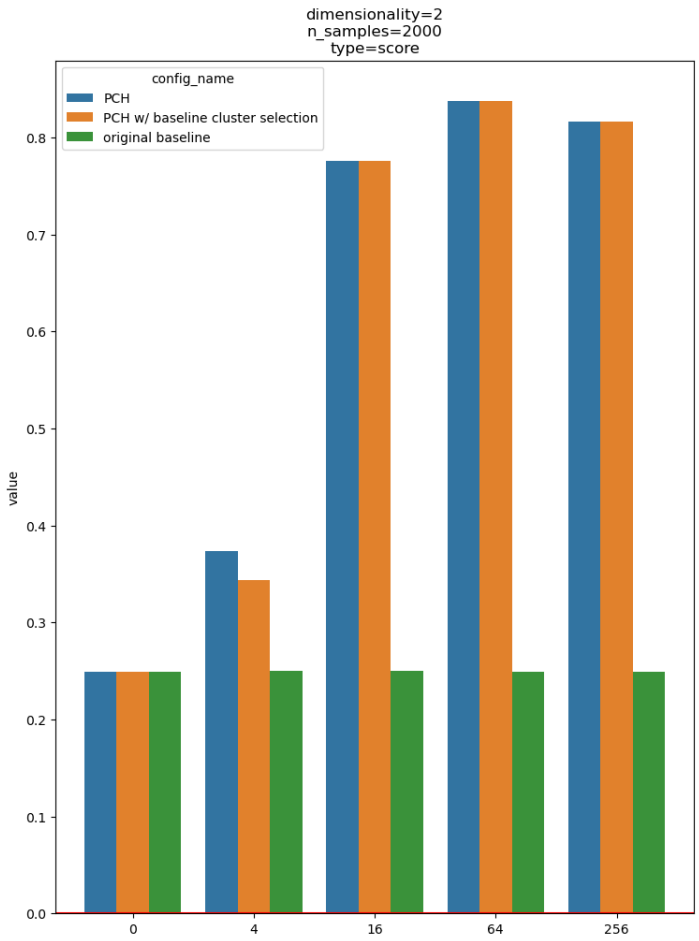


Figure 2.12: The ARI across several methods on a synthetic antagonistic dataset with spatial-locality expectation violations.

the features provided are quantitative analytical quantities from the wine composition, meaning there is a significantly subjective nonlinear process from features to labels.

Note that for non-synthetic datasets, we evaluate both PCH and the baseline algorithm multiple times for each number of constraints. Each run is stochastic in the choice of specific constraints, thus we attach an error bar to denote the $95\%$ confidence interval across the runs. All claims are made having verified the statistical significance of results using monte-carlo hypothesis testing, after adjusting p-values based on the Benjaminini-Yekutieli procedure to ensure that the false-discovery rate for statistical significance is fixed at $\alpha = 0.05$ despite multiple-hypothesis testing (in the sense that each amount of constraints serves as a separate hypothesis claiming significance) [6].

In addition to PCH and the HDBSCAN baseline algorithm, we also evaluate COP-KMeans as a representative of the KMeans family of semi-supervised algorithms. In the cases where COP-KMeans **fails to produce a clustering** due to the inability to reconcile new constraints with the enforcement of old constraints – a common theoretical problem – we take the result to be a score of zero, corresponding to the null-model of "random chance" labeling.

The data are pre-processed by projecting down to a two-dimensional space using an un-optimized UMAP projection with no minimum separation between points, allowing UMAP to optimized projected distances freely [55]. This projection strategy is the sole pre-processing done, and is repeated for each real-world dataset.

On this dataset, both PCH and the baseline algorithm gain in clustering performance, however with two interesting caveats: there seems to be an upper bound on how much the baseline method can benefit from additional constraints at some point, and the original baseline method has *reduced* constraint satisfaction as the number of constraints increases, whereas PCH only has a minor drop, as can be seen in figure 2.13. Furthermore, is competitive with COP-KMeans on low amounts of constraints, yet dominates at higher amounts. This is due to the fact that COP-KMeans is a greedy sequential algorithm with respect to constraints, which means that it is liable to arrive at irreconcilable states resulting in failed clusterings for larger numbers of constraints. This greatly limits its usability in general at all but the smallest scales.

### 2.6.4 Fashion-MNIST

Next we evaluate against the Fashion-MNIST dataset, known for its complexity and class ambiguity, making it a popular choice for validating clustering algorithms [91, 57]. In particular, Fashion-MNIST is a dataset of $28 \times 28$

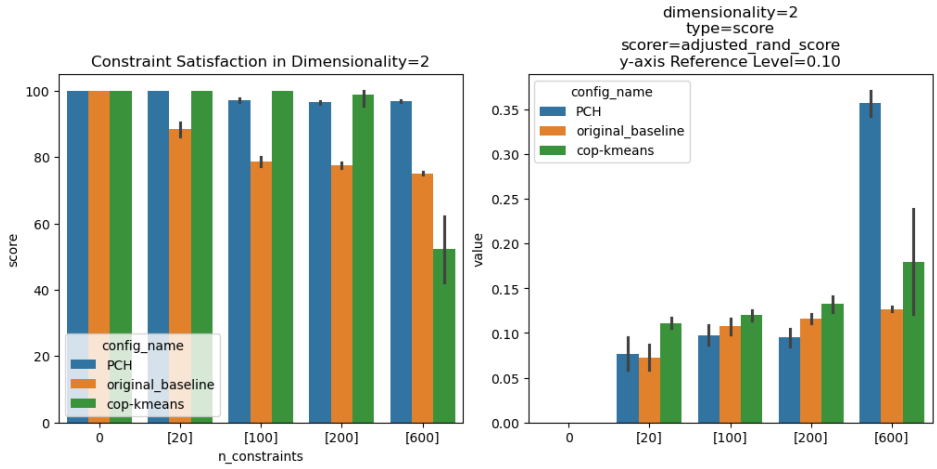


Figure 2.13: Constraint satisfaction and ARI of PCH, baseline semi-supervised HDBSCAN, and COP-KMeans algorithms on the Wine dataset.

images of articles of clothing, which serves as a great example of data with a high-dimensional original ambient space, significant complexity regarding relative feature interactions and correlations, along with a more complex (higher dimensional) data manifold. To estimate the data manifold, we project the dataset onto several different euclidean spaces and take the euclidean space with the highest vanilla HDBSCAN ARI as the closest match to the intrinsic dimensionality of the data manifold, under the assumption that the geometric nature of HDBSCAN would be most effective when the manifold is embedded in the same space, as opposed to being projected to lower or higher dimensions. For Fashion-MNIST, we project down to a 12-dimensional embedding space.

In figure 2.14 we can see that all models have a notably difficult time offering much gain over the traditional HDBSCAN model. This is largely due to the fact that different classes have incredibly high perceptual similarity and redundancy, to the point that their projected clusters overlap and intertwine, which fundamentally indicates that perhaps a different label schema would be more important. Consequently, there is an upper bound to how well the clusters can be separated in a fixed embedding space, and how well these clustering algorithms can derive a clustering that suits the semantic labels. That means that the vanilla HDBSCAN is already *near optimal*, which poses an especially interesting edge case. Due to its near optimality, we can see it initially *suffering* from having only a few constraints, whereas with sufficient constraints PCH is able to offer additional gains while the remaining two algorithms cannot.

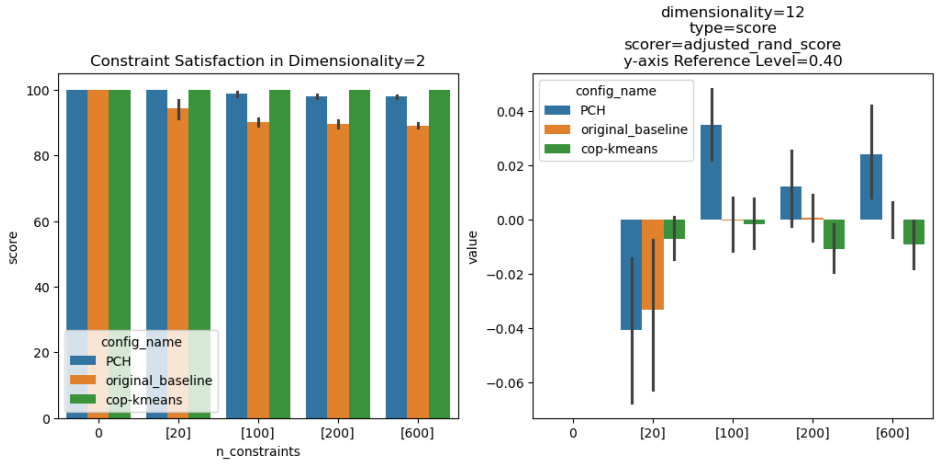


Figure 2.14: Constraint satisfaction and ARI of PCH, baseline semi-supervised HDBSCAN, and COP-KMeans algorithms on the Fashion-MNIST dataset.

### 2.6.5 Anuran Calls

Finally, we evaluate the algorithm on the relatively recently released Anuran Calls (AC) dataset [12]. This dataset is a fantastic example of situations where even *well-defined quantitative features* are not sufficient to make spatial locality in the representation space correspond directly to the intended semantics. The AC dataset consists of features derived from short, approximately 8 second anuran calls, with each sample being multi-labeled according to the family, genus, and species of the frog which generated the call. The nature of the labels as an intrinsically hierarchical taxonomy allows us to explore the role of PCH as a semi-supervised *hierarchical* clustering mechanism while also testing its ability to handle spatial-locality expectation violations, since animals that have similar taxonomies can present with wildly different phenotypes and consequently may form distinct clusters across the feature space, despite semantically being grouped together.

To that effect, we evaluate the original HDBSCAN baseline algorithm, PCH, and COP-KMeans across the AC dataset at all three available taxonomic levels of family, genus, and species in figures 2.15, 2.16, and 2.17 respectively. We can readily observe that PCH and the original baseline model tend to perform better at greater levels of specificity, peaking at the species level, meanwhile COP-KMeans performs consistently (poorly) across all three scales, largely due to its capacity as a greedy global optimizer. Meanwhile, the original baseline method struggles to make use of the new information presented by constraints,

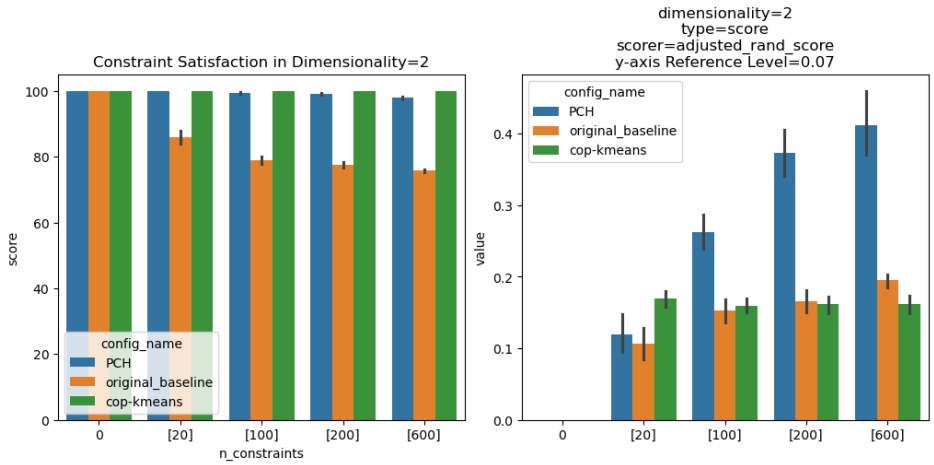


Figure 2.15: Constraint satisfaction and ARI of PCH, baseline semi-supervised HDBSCAN, and COP-KMeans algorithms taken across the AC dataset at the family level.

since it is limited by the original geometric hierarchy of HDBSCAN, resulting in decreased constraint satisfaction even when there are gains in ARI. Unlike the prior methods, PCH is able to significantly improve the ARI and scales well with the number of constraints, while maintaining constraint satisfaction.

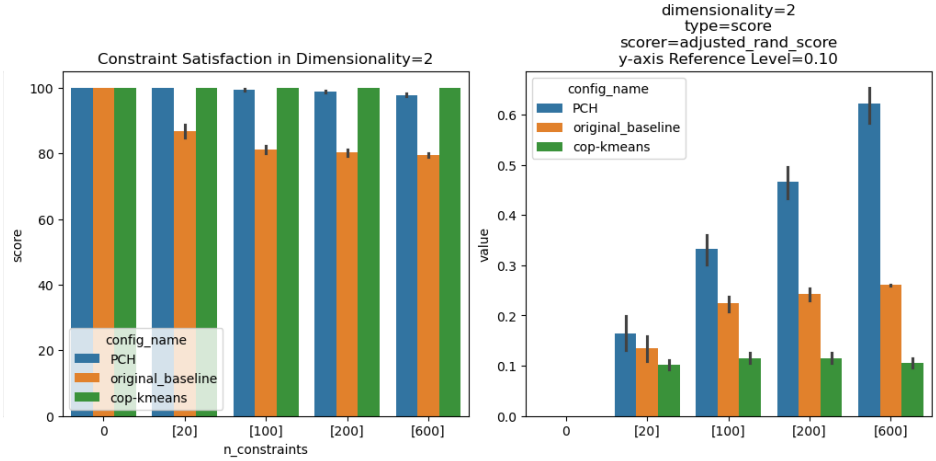


Figure 2.16: Constraint satisfaction and ARI of PCH, baseline semi-supervised HDBSCAN, and COP-KMeans algorithms taken across the AC dataset at the genus level.

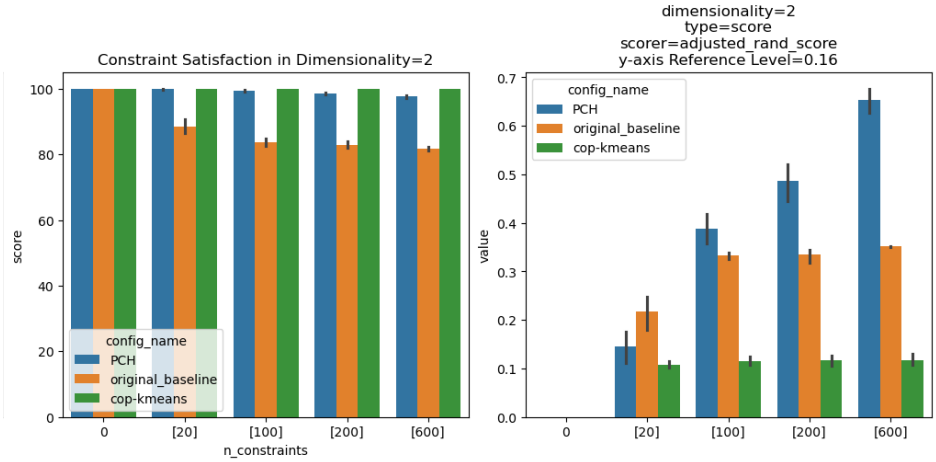


Figure 2.17: Constraint satisfaction and ARI of PCH, baseline semi-supervised HDBSCAN, and COP-KMeans algorithms taken across the AC dataset at the species level.

# Chapter 3

# Learned Data Introspection

## 3.1 Introduction

We develop PCH as a mechanism to allow for semi-supervised hierarchical clustering on arbitrary representation spaces in a way that prioritizes constraint satisfaction and expands the solution space. The expanded solution space is able to include clusterings which may be potentially non-local and contrary to spatial bias. While these goals are demonstrably achieved, we further augment the framework by inspecting the representation space itself and determining whether there is more to be done there. In particular, how does dimensionality reduction affect the downstream clustering task? Generally, both clustering and generative modeling algorithms often perform poorly on ambient-space data due to their high dimensionality. While there exist many dimensionality-reducing techniques, they exhibit various tradeoffs regarding the balance between preservation of local and global structure, computational costs, representation complexity and its consequences, etc. In this section, we will discuss the general problem of dimensionality reduction, discuss our preferred method of dimensionality reduction, as well as introduce Local Two-Stage Variational Autoencoder (LST-VAE), a novel architecture for dimensionality reduction that optimizes for a cluster-friendly structure in its transformation while specifically preserving the ability to produce arbitrary measure on the data manifold, which is a property often lost in dimensionality-reduction techniques.

## 3.2 Related Works

Dimensionality reduction is generally accomplished by either explicit feature selection, which aims to retain only the most informative and valuable features for the task at hand while eliding others, or transformation of inputs into lower-dimensional outputs based on various methods [76]. Going forward, we refer only to this latter process as dimensionality reduction, disentangling the notion of feature selection. Much of the modern data science workload involves complex, raw input data such as images, text, recordings, and other unprocessed media, all data formats lacking singular, highly informative features, which are generally an outcome of post-processing on all but the simplest data modalities [38]. The absence of these features makes dimensionality reduction a far more effective approach than feature selection for such input data. Dimensionality reduction can be further divided into families of linear and nonlinear methods. Linear methods are relatively constrained yet offer computationally efficient means of acquiring geometrically simple, interpretable results. In contrast, nonlinear methods are often more powerful, offering a wider solution space at the expense of interpretability and efficiency [24].

Starting with conventional linear models, Cunningham and Ghahrmani describe them as "program[s] with a problem-specific objective over orthogonal or unconstrained matrices," [17] noting the description's applicability to popular linear methods such as principal component analysis (PCA) [61, 20], multidimensional scaling (MDS) [85, 16], Fisher's linear discriminant analysis (LDA) [23, 63], factor analysis (FA) [77], distance metric learning (DML) [43, 94], canonical correlations analysis (CCA) [34], sufficient dimensionality reduction (SDR) [26, 1], etc. In particular, they formulate all of these methods, as well as several others, as a generalized optimization problem of finding $M \in \mathcal{M}$ to minimize an objective function $f_X(M)$ parameterized by the data. As a result, linear dimensionality reduction resolves down to optimizing a data-parameterized objective function over a specific matrix manifold. The solution space for such methods is thus determined by the type of matrix manifold and the type of optimization function. Often the matrix manifold will either be an arbitrary, unconstrained rank $r$ mapping or a pseudo-orthogonality constraint where $M^\intercal M = I$, leaving the objective function to further determine the solution space [17].

As mentioned earlier, while linear models come with many benefits including relatively simple interpretations, straightforward application, and generally efficient computation, they ultimately lack complexity. As formulated by Cunningham and Gharmani, the reliance of optimization over two particular

families of matrix manifolds results in a limited solution space that is only further specialized by their objective function. While no method is truly universal, nonlinear methods may optimize over wider domains and may express transformations and reductions that do not abide by linearity and thus are intrinsically unattainable by linear methods.

Nonlinear dimensionality reduction techniques include manifold methods, such as Local Linear Embeddings [68], Isomap [78], Laplacian eigenmaps [5], diffusion maps [15], t-distributed stochastic neighbor embedding (t-SNE) [51], and maximum variance unfolding [88], which construct nonlinear manifolds through local methods. In particular, they often rely on local neighborhoods, connectivity, geodesic distances, and other forms of graph-based local geometry to generate a manifold which is both locally consistent and optimal, often at the expense of global structure. These methods are often ill-conditioned when mapping pseudo-inverse transformations back into the data space, and do not provide a linear mapping to the lower-dimensional manifold. Despite some of these methods relying on eigenvalue problems and solutions and matrix factorization, their construction results in a necessarily nonlinear transformation. In general, many matrix-factorization methods such as nonnegative matrix factorization (NMF) [45] seem as though they ought to be linear models, yet the transformations they produce cannot be represented as linear maps and thus join the ranks of other nonlinear methods. An especially interesting nonlinear dimensionality reduction technique is uniform manifold approximation and projection (UMAP) [55]. This algorithm is SOTA and has very few assumptions. It assumes that the data is uniformly distributed on a Riemannian manifold, consistent with the manifold hypothesis, that the manifold is locally connected, and that the Riemannian metric can be approximated as locally constant. The first and final assumptions are common within the manifold hypothesis, and indeed we utilized the same assumptions earlier in both HDBSCAN and PCH. The very act of calculating MRDs is an approximation of the locally-constant Riemannian metric.

Recent development has highlighted the potential for autoencoders in dimensionality reduction, in particular emphasizing the utility of Variational Autoencoders (VAEs) [41]. With the rise of deep learning, especially generative modeling, VAEs have been a go-to strategy for dimensionality reduction in large models like large language models (LLMs) [56] and modern diffusion-based image generation models [65]. This is owed largely to the fact that VAEs are able to encode arbitrary information streams without explicit guidance on the structure of the information by using pseudo-invertible transformations, allowing a pseudo-bijection between the ambient data space and the learned represen-

tation space [42, 82]. Recent developments have significantly improved VAE performance at large, especially with novel perspective on geometric interpretations of the VAE latent space [2, 18].

As alluded to before, linear methods often struggle in handling more complex, high-dimensional, "natural" data such as images, videos, and other modalities. In particular, they are often unable to discern especially small local subspaces within the data, faring better at ascertaining macroscopic tendencies and patterns across the data [3]. Turning to nonlinear methods, UMAP is an especially compelling one since it generally produces high-quality transformations which preserve local connectivity much like t-SNE and other nonlinear embeddings, but also preserves global structure much more strongly, making it usable and reliable as a pre-processing step, much like PCA and much *unlike* t-SNE. Despite preserving semblances of both local and global structure, UMAP does not preserve the distribution of points in the sense that there are no guarantees regarding the density of points in the embedding, aside from a minimum distance hyperparameter which limits how close points may be embedded to each other.

Preservation of density, or at least the ability to explicitly model the density distribution of data is an important feature for clustering purposes since many clustering algorithms rely on underlying assumptions regarding the density of the data. For example, Gaussian mixture models (GMMs) assume normalcy of data under individual clusters, and an aggregate form of several clusters in a uniform mixture (though uniformity is alleviated when using weights on the GMM). While HDBSCAN and PCH do not have explicit density assumptions, empirically they work best when clusters are relatively uniform in density, while they grow sparse at their boundaries.

## 3.3 Background

### 3.3.1 Variational Autoencoders

In order to obtain precise control over the analytical distribution of data, we turn to VAEs which allow us to encode both a prior expectation over how we want the data to be distributed in its reduced-dimensionality form, as well as an explicit set of pseudo-inverse maps between the ambient and latent spaces. In particular, a VAE can be understood as a simple generative model with a few key components. Given a dataset $X = \{x_i\}_i^N \subset \mathbb{R}^n$ of $N$ i.i.d. samples of some random variable $\mathcal{X}$ (which may be discrete, or continuous), we assume that the data follow a two-step generative process defined by $p_{\theta^*}(z)p_{\theta^*}(x|z)$.

This process first samples a point in some latent space determined by a prior distribution $z \sim p_{\theta^*}(z)$ which is then used to generate a conditional distribution in the data space, from which a sample is finally drawn $x \sim p_{\theta^*}(x|z)$. Thus we define the *encoder* $q_\theta(z|x)$, the *prior* $p_\theta(z)$, and the *decoder* $p_\theta(x|z)$, where a subscript of $\theta$ denotes being parameterized by jointly learnable weights, in order to model this generative process. While the role of prior and posterior components, $p_\theta(z), p_\theta(x|z)$ respectively, are obvious since they have corresponding components in the generative process, the role of the $q_\theta(z|x)$ distribution is a bit more subtle, and is necessary to dispel the apparent intractability of the problem. Generally these components are assumed to come from the same parametric family of distributions, however that need not always be the case. For simplicity, the original VAE will be defined with each component as a parameterized Gaussian distribution. In particular, $p_\theta(z),\ q_\theta(z|x)$ are distributions defined over a lower-dimensional embedding space, known as the latent space, generally $\mathbb{R}^d$ for some $d << n$.

The optimization target of a VAE is to maximize the likelihood of the data under the model, i.e. maximize $p_\theta(x)$. Following the generative process, this means maximizing $\int p_\theta(z)p_\theta(x|z)dz$ which is unfortunately intractable due to the need to integrate over the entire latent space, thereby making it impossible to differentiate. Moreover, we have that the true posterior density $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$ itself is also intractable, meaning that many expectation-maximizing and variational methods cannot be used [41]. To remedy this issue, Kingma and Welling consider the variational lower bound on the marginal likelihood. Specifically

$$\ln p_\theta(x_1, \ldots, x_N) = \sum_{i=1}^{N} \ln(p_\theta(x_i))$$

which can be rewritten by noting that

$$\ln(p_\theta(x_i)) = D_{KL}(q_\theta(z|x_i)||p_\theta(z|x_i)) + \mathbb{E}_{q_\theta(z|x)}[-\ln q_\theta(z|x) + \ln p_\theta(x, z)]$$

where the first term on the RHS is the Kullback-Leibler divergence [44] between the approximate and true latent-code posterior and the second term is the so-called variational lower bound. We may denote the variational lower bound as

$$\mathcal{L}(\theta; x_i) = \mathbb{E}_{q_\theta(z|x)}[-\ln q_\theta(z|x) + \ln p_\theta(x, z)]$$

noting that $\mathcal{L}(\theta; x_i) \leq \ln p_\theta(x_i)$. Thus, maximizing the variational lower bound serves as optimizing the lower bound of the marginal likelihood, providing an indirect mechanism to optimize the system for the generative process.

This is made much more doable by further decomposing

$$\mathcal{L}(\theta; x_i) = -D_{KL}(q_\theta(z|x_i)||p_\theta(z)) + \mathbb{E}_{q_\theta(z|x_i)}[\ln(p_\theta(x_i|z))]$$

which completes our optimization target.

Then, we then may revisit and concretely define the encoding process which is used to map a data point $x$ to a latent code $z \sim q_\theta(z|x)$. Since we have that $q_\theta(z|x)$ is some Gaussian, we may denote its parameterization as

$$q_\theta(z|x) = \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x)) = \mathcal{N}(\mu_\theta(x), \sigma_\theta(x))$$

where for simplicity we assert that $\Sigma_\theta$ is a diagonal covariance matrix, denoting it as $\sigma_\theta$ instead. After encoding, we define decoding as the map from $z$ to

$$\hat{x} = p_\theta(x|z) = \mathcal{N}(\mu_\theta^x(z), \sigma_\theta^x(z)) = \mathcal{N}(\mu_\theta^x(z), I)$$

where generally we fix $\sigma_\theta^x = I$ for convenience.

Finally, the prior

$$p_\theta(z) = \mathcal{N}(\mu_\theta^p, \sigma_\theta^p) = \mathcal{N}(0, I)$$

is defined as a target such that the VAE learns to produce latent distributions that follow the distribution of the prior, where for convenience, we generally take $\mu_\theta^p = 0,\ \sigma_\theta^p = I$.

Note that since $z := \mu + \epsilon_n \Sigma$ for $\mu \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n}$ and $\epsilon_n \sim \mathcal{N}(0, I_n)$ is distributed as $z \sim \mathcal{N}(\mu, \Sigma)$, we can further rewrite $q_\theta(z|x) = \mu_\theta(x) + \epsilon_d \cdot \sigma_\theta(x)$ and $p_\theta(x|z) = \mu_\theta^x(z) + \epsilon_n$ where we take the element-wise product with $\epsilon$, allowing us to directly take the gradient with respect to the parameters $\mu_\theta, \mu_\theta^x, \sigma_\theta$.

Since we are parameterizing the distributions through choices of $\mu,\ \sigma$, we can rephrase the construction of a VAE into that of a neural network directly providing these parameters. Specifically, we define the encoder $f : \mathbb{R}^n \to \mathbb{R}^{2 \times d}$ that maps $x \to \mu_\theta(x), \sigma_\theta(x)$ providing the parameterization for $q_\theta(z|x)$. Similarly, we define the decoder $g : \mathbb{R}^d \to \mathbb{R}^n$ that maps $z \to \mu_\theta^x(z)$ providing the parameterization for $q_\theta(z|x)$.

In which case, our optimization target can be rewritten noting that the KL-divergence term may be rewritten as

$$-D_{KL}(q_\theta(z|x_i)||p_\theta(z)) = \frac{1}{2}\sum_{j=1}^{d}(1 + \ln(\sigma_\theta(x_i)_j^2) - \mu_\theta(x_i)_j^2 - \sigma_\theta(x_i)_j^2)$$

and the marginal log-likelihood term can be rewritten as a monte-carlo estimate (since our samples follow $p(x)$ and consequently our latent codes follow $q_\theta(z|x)$). We denote likelihood of a point $x$ under $\mathcal{N}(\mu, \sigma)$ as $f(x; \mu, \sigma)$:

$$\begin{aligned}
\mathbb{E}_{q_\theta(z|x_i)}[\ln(p_\theta(x_i|z)] &= \ln\left(\prod_{j=1}^{n} f((x_i)_j; (\mu_\theta^x)_j, (\sigma_\theta^x)_j)\right) \\
&= \sum_{j=1}^{n} \ln(f((x_i)_j; (\mu_\theta^x)_j, (\sigma_\theta^x)_j)) \\
&= -\frac{1}{2}\left(n\ln(2\pi) + \sum_{j=1}^{n}\left(2\ln((\sigma_\theta^x)_j) + \frac{((x_i)_j - (\mu_\theta^x)_j)^2}{(\sigma_\theta^x)^2}\right)\right)
\end{aligned} \tag{3.1}$$

which due to our choice of $\sigma_\theta^x = I$, and dropping the constant term, simplifies to

$$-\frac{1}{2}\sum_{j=1}^{n}\left(((x_i)_j - (\mu_\theta^x)_j)^2\right)$$

Putting it all together, our case of a simple VAE optimizes the final objective function[8] of

$$\begin{aligned}
-\mathcal{L}(\theta; x_i) = &\frac{1}{2}\sum_{j=1}^{n}\left(((x_i)_j - (\mu_\theta^x)_j)^2\right) \\
&- \frac{1}{2}\sum_{j=1}^{d}(1 + \ln(\sigma_\theta(x_i)_j^2) - \mu_\theta(x_i)_j^2 - \sigma_\theta(x_i)_j^2)
\end{aligned} \tag{3.2}$$

[8] Generally the optimization target is framed as a minimization objective, so since we intend to maximize the variational lower bound, in the objective function we minimize the *negative* of the bound itself.

While this describes the basics of a VAE, as formulated by Kingma and Welling [41], there are myriad modifications which build atop this framework, usually by finding unique implementations of $q_\theta(z|x)$, $p_\theta(z)$, $p_\theta(x|z)$ or more rarely by altering the optimization target and optimization process itself [82, 18, 102, 75, 2]. In the next two sections, we will explore two such modifications.

### 3.3.2 VampPrior

A noted empirical limitation of VAEs is that setting a unit Gaussian as the prior tends to be restrictive, making it more difficult for the VAE to learn arbitrary distributions and distinguish clusters within the dataset while also serving as a suboptimal target from a theoretical perspective, forcing *more work* to be done in the first place [2, 82]. It is worth noting that such a prior is restrictive *only* in the empirical sense, since a Gaussian prior does not reduce the solution space of

a VAE when its latent dimension matches that of the underlying data manifold [18]. Regardless, much work has been done in pursuit of a flexible, robust, interpretable choice of prior that streamlines training [102, 64, 81, 82]. In a sense, we can solve for an *ideal* prior based on the premise of maximizing the variational lower bound, and indeed this is what Tomczak and Welling do in "VAE with a VampPrior" [82]. They cleverly consider finding an optimal prior by framing it as optimizing for the prior which results in the greatest log-likelihood under the aggregate posterior distribution in the latent space (that is, the aggregate distribution achieved by mapping the dataset through the encoder). This optimization problem can be written simply as maximizing the following Lagrange function, with Lagrange multiplier $\beta$:

$$\max_{p_\theta(z)} -\mathbb{E}_{z\sim q(z)}[-\ln(p_\theta(z))] + \beta\left(\int p_\theta(z)dz - 1\right)$$

where the final term is simply the Lagrange multiplier times the constraint that the function being found is indeed a probability distribution. Almost trivially, the solution to this optimization is just the aggregate posterior itself:

$$p_\theta(z) = \frac{1}{N}\sum_{i=1}^{N} q_\theta(z|x_i)$$

Despite its simplicity, this solution has its own complexities. It is known that such a distribution would likely overfit the model to the data, and a full aggregate posterior is incredibly expensive to compute per-batch, making the distribution near-intractable for most data [32, 52, 82].

Instead, we can *approximate* the aggregate prior by including $M$ learnable tensors $\{x_i^\eta\}_{i=1}^M$ in our model. These new tensors, called pseudo-inputs, take the shape of the input data and exist to be mapped to posterior latent distributions in order to generate an aggregate distribution to serve as our approximate prior:

$$p_\theta(z) = \frac{1}{M}\sum_{i=1}^{M} q_\theta(z|x_i^\eta)$$

We use this approximate prior to compute the KL-divergence term in the variational lower bound. Thus, the gradient can directly pass through and optimize the pseudo-inputs so that they best serve as approximate samples for the true aggregate prior. It is worth noting that in practice, due to the curse of dimensionality, pseudo-inputs will often "look" like random noise more than actual data. This is because, in the case of the manifold hypothesis, the data manifold has volume zero (even though its noisy expansion has non-zero vol-

ume), thus it is unlikely (but not impossible) for a pseudo-input to be learned on the data manifold (or its noisy expansion) itself. We discuss this further in a later section.

### 3.3.3 Two-Stage VAE

While conventional wisdom has widely accepted that indeed limited choices of prior and posterior distributions can make it difficult to produce good results in practice, this need not be the case. In particular, Dai and Wipf observe that in the case where the manifold dimensionality matches the ambient space ($n = d$), even a simple VAE can achieve a globally optimal solution under the standard Gaussian parameterizations.

**Definition.** A $k$-simple VAE is a VAE with latent dimension $d = k$ whose encoder and decoder, are both parameterized Gaussian distributions. In particular, we define $p_\theta(x|z) = \mathcal{N}(\mu_\theta^x, \sigma_\theta^x)$ and, unlike in our earlier discussion of basic VAEs, we do not set $\sigma_\theta^x = I$. Instead, we set $\sigma_\theta^x = \gamma I$ for a learnable scalar parameter $\gamma$.

Dai and Wipf prove that there exists a sequence of parameterization of a $k$-simple VAE, with parameters denoted $\theta_t^*$ such that

$$\lim_{t\to\infty} D_{KL}(q_{\theta_t^*}(z|x)||p_{\theta_t^*}(z|x)) = 0$$

and

$$p_{\theta_t^*}(x) \to p(x) \textit{ almost everywhere}$$

which is equivalent to an "ideal" VAE distribution, *despite* the apparent limitations of the choice of prior and posteriors. Please refer to [18] for precise details on the proof. In the case of ambient space dimension greater than the manifold dimension ($d < n$, as is almost always the case) then the situation is significantly more nuanced. Let $\lambda_{gt}$ be the "ground-truth" measure on the underlying data manifold $\mathcal{M}$ such that our ideal $p_\theta(x)$ is actually described by the measure $\lambda_{gt}$. In this case, Dai and Wipf similarly show that

$$\lim_{t\to\infty} D_{KL}(q_{\theta_t^*}(z|x)||p_{\theta_t^*}(z|x)) = 0$$

and

$$\lim_{t\to\infty} \int_{\mathcal{M}} -\ln(p_{\theta_t^*}(x)) d\lambda_{gt} = -\infty$$

while

$$\forall A \subset \mathbb{R}^n \mid \lambda_{gt}(\partial A \cap \mathcal{M}) = 0, \lim_{t\to\infty} \int_{x\in A} p_{\theta_t^*}(x)dx = \lambda_{gt}(A \cap \mathcal{M})$$

Note that there is a similarity with the last case, in that the KL-divergence can be pushed towards zero reliably. The difference is that while the former case demonstrated that the parameterized distribution can arbitrarily approximate the true distribution, here we have a more nuanced version of that: the third equation can be interpreted as saying that the same KL-minimizing sequence can match the ground-truth distribution everywhere on the data manifold[9] while still observing equation two, which is that the distribution will drive the lower-bound *arbitrarily low*, chasing a potentially trivial solution to the optimization problem. Most importantly, the conditions of *optimizing the lower bound* and *functionally recovering the ground-truth measure* are distinct in this case, meaning that one may be pursued without the other.

[9] though technically it is on the volume-zero overlapping sets, however it is simpler and not-much-worse to understand it as the manifold itself

The main consequence of this fact is that the intrinsic limitations of a VAE are more to do with dimensionality than with choice of prior or posterior. Although they may have empirical effects, they are not necessarily bottlenecks as conventional wisdom suggests. Consequently, Dai and Wipf propose an implementation-ready solution to this limitation. Since a VAE can only be guaranteed to have the chance to reach an optimal solution when the manifold dimension and ambient dimension are equal, they divide up the VAE training process into two steps, across two different $k$-simple VAEs. First, they train a $k$-simple VAE on the dataset. Note that since their definition of a $k$-simple VAE includes a term for the generative posterior variance ($\gamma$), it observes Theorem 4 in [18], summarized as:

**Theorem.** *Let $\theta_\gamma^*$ denote the parameters of a $k$-simple VAE where the ambient dimension is greater than the manifold dimension ($k < n$) and $\gamma$ is the sole optimized parameter in the VAE. Then for any $\gamma > 0$, there exists $\gamma' < \gamma$ such that $\mathcal{L}(\theta_{\gamma'}^*) < \mathcal{L}(\theta_\gamma^*)$.*

Recall that the guarantee of an optimal solution for a VAE is predicated on the inclusion of $\gamma$ in the definition of a $k$-simple VAE. In particular, the presence of the $\gamma$ term allows for the decoder distribution to limit towards a fixed point-wise representation of the output. Semantically, this can be understood as optimizing for the *data manifold* (rather precisely) at the expense of a sense of the underlying *distribution* on that manifold. While you are likely to obtain a latent embedding yielding a low reconstruction error, it will likely not be distributed according to $p_\theta(z)$. To remedy this, we treat the latent embedding as its own ambient space and train the second $k$-simple VAE on it. Now we

can guarantee that the ambient dimension and manifold dimension are equal[10] which means that we can globally optimize the variational lower bound to learn both the manifold embedding (in this case an involution would suffice) *and* the distribution on the manifold.

[10] assuming an initial choice of manifold dimension small enough to match the data manifold dimensionality

With this formulation, we now have a concrete handle on the distribution of the first VAE. In particular, to sample $x \sim p_\theta(x)$ one must instead sample $u \sim p_\phi(u)$ from the second VAE, then take $z \sim p_\phi(z|u)$ and finally $x \sim p_\theta(x|z)$. This ancestral sampling allows us to utilize the first VAE as an optimal choice of decoder, while we leverage the second VAE as a complete prior by itself.

## 3.4 Methodology

### 3.4.1 Natural VampPrior

As we have established, a $k$-simple VAE is sufficient to obtain a global optimum in the case of same-dimension representation. However, empirical training rarely obtains such global optima, and intelligent, deliberate choices of prior distribution can help stabilize VAE training and produce representations that are more qualitatively useful regardless of quantitative optimality. To this end, we favor VampPrior as a choice of prior due to its ability to serve as a GMM while being entangled directly to the encoder as during optimization. This entanglement further stabilizes the VAEs training and ensures a smoother optimization [82]. Furthermore, the fact that pseudo-inputs take on the shape of the incoming data inspires hope for the notion that the pseudo-inputs will, in some way, be representative of the data. Unfortunately, in high-dimensional cases this is often *not* the case. We define a uniform random unit vector as $U = \frac{X}{\|X\|}$ where $X \sim \mathcal{N}(0, I_n)$. Consider the product of two uniform random unit vectors,

$$\langle U, V \rangle = \frac{\langle X, Y \rangle}{\|X\|\|Y\|}.$$

Since $\langle X, Y \rangle$ is ultimately the sum of $n$ variables $X_i Y_i = Z_i \sim W$ for some distribution $W$. Note that

$$\mathrm{Var}(Z_i) = \mathrm{Var}(X_i Y_i) = \mathrm{Var}(X_i)\,\mathrm{Var}(Y_i) = 1$$

Thus, we may apply the central limit theorem to obtain the fact that $\langle X, Y \rangle \xrightarrow{n\to\infty} \sqrt{n}\mathcal{N}(0,1)$. Note that by the law of large numbers, we have that

$$\frac{\|X\|}{\sqrt{n}}, \frac{\|Y\|}{\sqrt{n}} \xrightarrow{n\to\infty} 1$$

meaning that

$$\sqrt{n}\langle U, V \rangle \xrightarrow{n\to\infty} \mathcal{N}(0,1)$$

and therefore

$$\forall \epsilon > 0,\ P(|\langle U, V \rangle| > \epsilon) \xrightarrow{n\to\infty} 0$$

meaning that as the number of dimensions increase, the probability that two uniform random vectors are nearly orthogonal gets arbitrarily high. This result can be readily extended to note that the same is true for an arbitrary set of $k$ uniform random vectors, showing that a random set of high-dimensional vectors forms a quasi-orthogonal basis.

Since pseudo-inputs are initialized randomly in the ambient space, they themselves form an arbitrary quasi-orthogonal basis, often allowing the VAE to optimize them as "anchor points" that need not lie on the data manifold itself. Indeed, the learning capacity of the encoder means that it can take those finitely-many pseudo-inputs and embed them into the latent space in convenient positions that would imply they are on-manifold, while still being random noise in the ambient space. As previously discussed in the context of two-stage VAEs, arbitrary optimization may not lead to proper ground-truth aligned distributions. This behavior limits the ability of pseudo-inputs to be optimized in a way that allows them to resemble actual data. This is further complicated by the fact that there exists a degenerative local minimum to the KL-divergence term of the variational lower bound for a VampPrior: if each pseudo-input converges to the average of all pseudo-inputs, this allows input encodings to optimize for similarity to a *single* Gaussian prior instead of a mixture of multiple distinct Gaussians. Though this equilibrium may result in lower computed loss, it generally reduces the model to a regular VAE with a Gaussian prior.

To alleviate both of these issues, we introduce Natural VampPrior (NVP) which modifies the VampPrior strategy in three key ways:

1. We first initialize the pseudo-inputs based on real data sampled randomly from the dataset.

2. We *freeze* the pseudo-inputs during the first stage of VAE training, and re-train them *separately* while keeping the rest of the model frozen.

3. We add an additional regularization term to the loss, which is a parallel of the usual variational lower bound except as calculated for the pseudo-inputs.

The first item helps ensure that the pseudo-inputs are, *in aggregate*, representative of the data and provide a reasonable cover (which becomes more true the more pseudo-inputs are used). The second item instead helps ensure that we avoid the attractive local optima of homogeneous pseudo-inputs in the first phase of training. The second phase can be understood as a sort of "projection" of the pseudo-inputs back onto the data manifold itself in a way that matches the ground-truth distribution. This is guaranteed by the third item, which sees us add the following term to the loss as introduced in [98]:

$$\eta\mathcal{L}^{\eta}(\theta; x_i) = \eta\left[-D_{KL}(q_\theta(z|x_i^\eta)||p_\theta(z)) + \mathbb{E}_{q_\theta(z|x_i^\eta)}[\ln(p_\theta(x_i^\eta|z)]\right]$$

which is equivalent to the variational lower bound applied with the pseudo-inputs as the data with which to condition the distributions on, as opposed to real data. The scaling factor of $\eta$ controls the balance between the pseudo-inputs and regular data in the final loss term. This term encourages that in the second phase where only pseudo-inputs are optimized, they learn to get arbitrarily close to the data manifold in the ambient space. The precision of such a projection is limited by the capacity and training of the network as a whole. An example can be seen in figure 3.4.

Another way to project the pseudo-inputs onto the data manifold would be to instead take the expected value of the latent distribution of the pseudo-inputs generative variables, $\bar{z}_i^\eta = \mathbb{E}[q_\theta(z|x_i^\eta)]$, and take the expected value of its conditional distribution in the data space, $\bar{x}_i^\eta = \mathbb{E}[p_\theta(x_i^\eta|\bar{z}_i^\eta)]$. This method fails in the case of a two-stage VAE due to the fact that while the projection can be reasonably guaranteed to land on the data manifold, due to the disentanglement of measure from the first-stage VAE, it may not end up at the right place on the manifold. This can be alleviated by altering the projection to using $\bar{u}_i^\eta = \mathbb{E}[q_\phi(u|z_i^\eta)q_\theta(z_i^\eta|x_i^\eta)]$ where $u$ is sampled from the second-stage VAE, $\bar{z}_i^\eta = \mathbb{E}[p_\phi(z|\bar{u}_i^\eta)]$ where now $\bar{z}_i^\eta$ is the expected value from the decoder of the second-stage VAE, while $\bar{x}_i^\eta$ is calculated analogously by running the first-stage decoder on $\bar{z}_i^\eta$.

Empirically we observe that the first method of re-training allows pseudos to better learn a composition that optimizes the KL-divergence of the posterior distributions, achieving in some sense a more "optimal" representation than the second method's projection would afford. The second method can only indirectly optimize based on the premise that an on-manifold pseudo-input would

exhibit lower loss than an off-manifold pseudo-input. We note this premise is not necessarily true in the case that the ambient dimension is greater than the manifold dimension.

Ultimately, either method yields pseudo-inputs that genuinely *look* like they came from the dataset (with fidelity depending on model capacity) and can be used as quick "representative" samples for the entire dataset. In particular, they will be representatives in areas of high density, resulting in a distilled snapshot of the variance across the data, allowing experts to distinguish how many of the pseudo-inputs represent novel and interesting cases as well as how many are redundant. The emphasis of distinct pseudo-inputs and the reduction of redundant pseudo-inputs are discussed further in the next chapter.

### 3.4.2 Pushforward Measure

While a traditional VAE optimizes both the geometric manifold and geometric distribution simultaneously, we have established that a two-stage $k$-simple VAE system is able to disentangle these aspects and represent them independently. This has many obvious benefits, including the general reconstruction ability of the first-stage geometric manifold learning VAE. However, there is a caveat: there is no longer a known distribution over the latent space that encodes the ground-truth distribution. The main consequence is that statistical quantities such as likelihood are no longer obtainable in the same way. Luckily, this can be easily remedied by considering the *pushforward measure*. Given a trained two-stage VAE system, we define a map $f : \mathbb{R}^n \to \mathbb{R}^d$ based on the *encoder* of the first stage, which takes $x \to \mu_\theta(x)$. Similarly we define $g : \mathbb{R}^d \to \mathbb{R}^d$, the *encoder* for the second-stage VAE where it takes $z \to \mu_\phi(z)$ where we use $\{\theta, \phi\}$ to denote the independent parameterizations of the two networks. Then, we define

$$\lambda_p := \lambda_d \circ g \circ f$$

where $\lambda_d$ is the $d$-dimensional measure on $\mathcal{M}$ learned by the second-stage VAE[11].

[11] Note that we need not concern ourselves with the behavior of the measure *off manifold* since by construction we take our manifold to be $\mathbb{R}^d$ and map *down* from the ambient space, thus the image of both encoders is within the manifold representation

### 3.4.3 Locally Specialized VAE

A natural generalization of VampPrior is to consider the weighted sum of learned pseudo-input posterior distributions with learnable weights $w$. We take this notion and develop it further in the hopes of aiding the separation of clusters in the latent space, allowing for smoother downstream clustering. First, we introduce a new hidden layer that operates one layer before the latent-dimension

bottleneck. This layer maps samples to an *evidence score* distribution which is used as logits to a softmax operation to determine how to weight the pseudo-input posteriors when calculating KL-divergence for *that sample* in particular. This allows samples to select which distributions they most identify with, allowing them to cluster more tightly and encouraging a spatially-disentangled relationship between pseudo-input posteriors. Specifically, we define $W : \mathbb{R}^l \to \mathbb{R}^K$ a linear map where $K$ is the number of pseudo-inputs, and we set $w = \mathrm{softmax}(XW)$ as the sample-wise weights where $X \in \mathbb{R}^{n \times l}$ is the matrix of samples.

By itself, this mechanism helps with the disentanglement of pseudo-inputs and their distributions. To further separate them, we introduce a stochastic map $M$ that samples from a parameterized categorical distribution of $K$ categories, $C = \mathrm{Cat}(K)$. Specifically, $M$ maps $x_i \to c_i \sim C(\mathrm{softmax}(x_i W))$ where the parameterization $c_i \sim C(\mathrm{softmax}(x_i W))$ refers to $\mathrm{softmax}(x_i W)$ acting as a list of probabilities for each category. Thus the categorical outcome of $M(x_i)$ determines *which* pseudo-input is used when calculating the KL-divergence. In this way we are able to isolate individual gradients to each sample and ensure that they flow with respect to *a single pseudo-input* at a time. We avoid affecting the overall stability of the training process since the stochastic nature the map paired with the many iterations of batch-wise training effectively enable us to take a monte-carlo estimate of the expected value of the mean distributions of the categorical maps.

We furthermore introduce a regularization term $\alpha$ to improve the entropy profile of $M$. Specifically, we want as many pseudo-inputs to be used as possible, meaning that across a batch we observe high entropy in $M$. Meanwhile, we want each sample to *decisively* choose its own pseudo-input, meaning for each distribution $M(x_i)$ we want to observe low entropy. Thus, taking $p_{i,j} = P_{M(x_i)}(c_i = j)$ and $p_j = \frac{1}{N} \sum_{i=1}^{N} p_{i,j}$ we define

$$\alpha = \left( \sum_{j=1}^{K} (p_j) \ln((p_j)) \right) - \frac{1}{N} \left( \sum_{i=1}^{N} \sum_{j=1}^{K} p_{i,j} \ln(p_{i,j}) \right)$$

The first term on the RHS is the batch-wise entropy, which is the entropy of the aggregate expected value of the categorical distributions. When batch-wise entropy is high, it means that many pseudo-inputs are being regularly used. The second term on the RHS is the average sample-wise entropy. When average sample-wise entropy is low, it means that the categorical distributions of each sample (i.e. $M(x_i)$ are tighter and closer to a decisive choice than a uniform random one. In particular, the gap in entropy quantities – which is $\alpha$ itself –

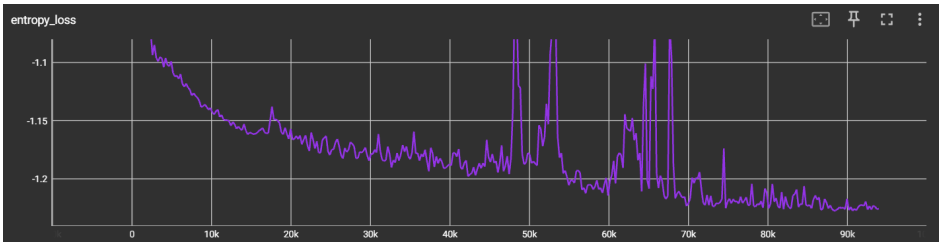


Figure 3.1: The value of $\alpha$, or the "entropy-gap" over the course of training.

represents how much of the population of pseudo-inputs does an average point draw from. In particular if we write $\alpha = -\mathbb{H}_b + \mathbb{H}_s$ where $\mathbb{H}_b, \mathbb{H}_s$ are batch-wise and sample-wise entropy respectively, then each sample draws from $e^\alpha$. Note that we can rewrite

$$\begin{aligned}
\alpha &= \sum_{j=1}^{K} \mathbb{E}[p_{i,j}] \ln(\mathbb{E}[p_{i,j}])) + \mathbb{E}\left[-\sum_{j=1}^{K} p_{i,j} \ln(p_{i,j}))\right] \\
&= -\left[-\sum_{j=1}^{K} \mathbb{E}[p_{i,j}] \ln(\mathbb{E}[p_{i,j}])\right] + \mathbb{E}\left[-\sum_{j=1}^{K} p_{i,j} \ln(p_{i,j}))\right] \\
&= \mathbb{E}\left[-\sum_{j=1}^{K} p_{i,j} \ln(p_{i,j}))\right] - \left[-\sum_{j=1}^{K} \mathbb{E}[p_{i,j}] \ln(\mathbb{E}[p_{i,j}])\right] \\
&= \mathbb{E}_i[\mathbb{H}_j[p_{i,j}]] - \mathbb{H}_j[\mathbb{E}_i[p_{i,j}]]
\end{aligned} \tag{3.3}$$

where

$$\mathbb{H}[p_j] = -\sum_i p_j \ln(p_j)$$

is the entropy over a discrete distribution with probabilities $p_j$. Note that since $x \ln x$ is concave-up, Jensen's inequality guarantees that $\alpha < 0$. We can see an example of this entropy gap in figure 3.1.

Since we separate the generative process into a hierarchical process with the two-stage VAE framework, we retain the disentanglement of the manifold-learning and measure-learning by implementing these structural changes in the *second* stage VAE, thus only the portion in charge of measure-learning need worry about the distribution of points and categorical clusters.

## 3.5 Results

We measure the efficacy of NVP by considering whether it is able to maintain or make gains on its loss optimization compared to a vanilla VAE, whether its representation space is more suitable for clustering, and whether it produces reasonable pseudo-inputs that can help with introspection. The first two measures are quantitative, while the last is qualitative and subjective. We perform our experiments on the Fashion-MNIST dataset using a lightweight residual convolution architecture consisting of approximately one million parameters, and limit our representation to two dimensions for computational simplicity [29]. As we can see in figure 3.5, the training regimes for both models are comparable yet the NVP model converges to a significantly lower loss than the vanilla VAE. The adoption of pseudo-inputs *increases* the general performance of the VAE network, and the use of NVP loss and training regime preserves those gains [82].

Next, we compare the learned pseudo-inputs to the actual sample data. Please see figure 3.3 for sample data, and figure 3.4 for an example of learned pseudo-inputs. Note that the pseudo-inputs generally match the sample distribution in terms of general composition (e.g. many shoes and tops, few purses). While pseudo-inputs are not as detailed as the true sample data, this is a limitation of the capacity and training of the neural network used in the experiment rather than an intrinsic flaw of the approach itself. Despite the lack of fine details, the pseudo-inputs themselves already serve to offer a few easy partitions among the data for an investigating expert. The composition of the pseudo-inputs suggests that we can roughly form the following low-fidelity groups: shirts, pants, purses, and shoes. Investigating further (perhaps with additional pseudo-inputs) could then suggest stratifying the classes so as to distinguish flat shoes from heels and ankle-boots, and perhaps fragmenting the "pants" class in which pants and dresses often co-cluster into one class of true pants and another class of dresses.

Zooming out, we can see in figure 3.7 that the pseudo-inputs provide a relatively uniform cover of the empirical distribution in the embedding space. While pseudo-inputs don't correspond 1-1 to the canonical labels for the dataset, they tend to correspond to *perceptually similar* groups (e.g. long pants and dresses) and provide a separate and unique perspective on what the "right" clustering on such a dataset would look like.

Next we consider the locally specialized VAE (LSV) model which builds on top of NVP. We compare it directly against NVP in the same context as the prior comparison between NVP and a vanilla VAE. Despite the implementation

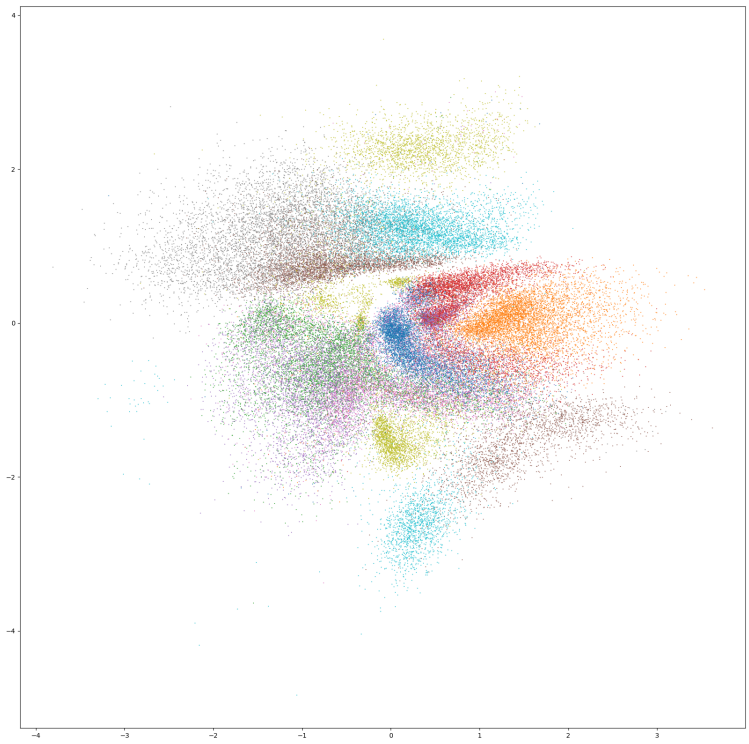


Figure 3.2: The embedding space of a trained VAE model. Points are colored by their ground-truth labels under the Fashion-MNIST dataset.

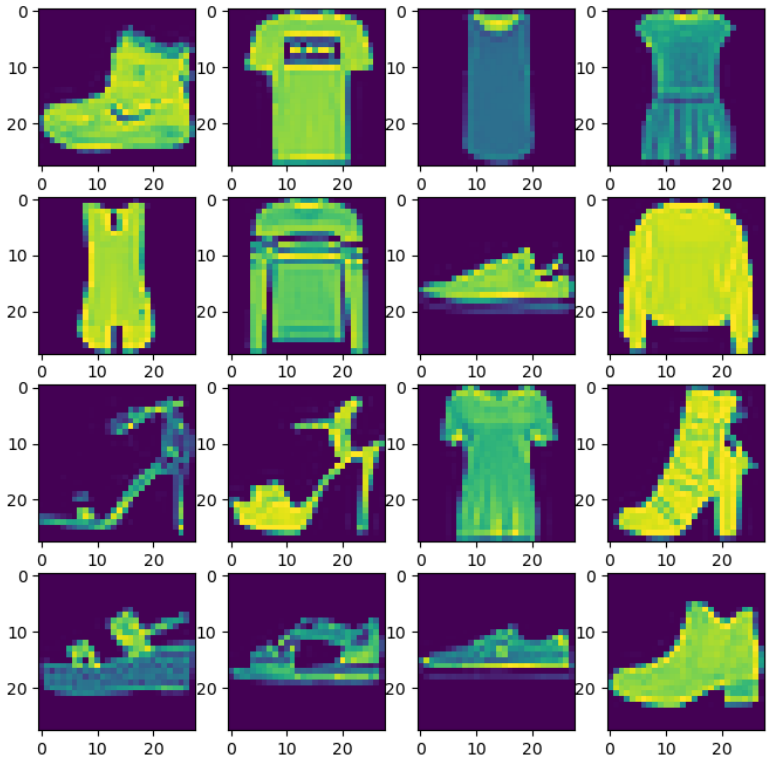

Figure 3.3: Sixteen samples drawn from the Fashion-MNIST dataset.

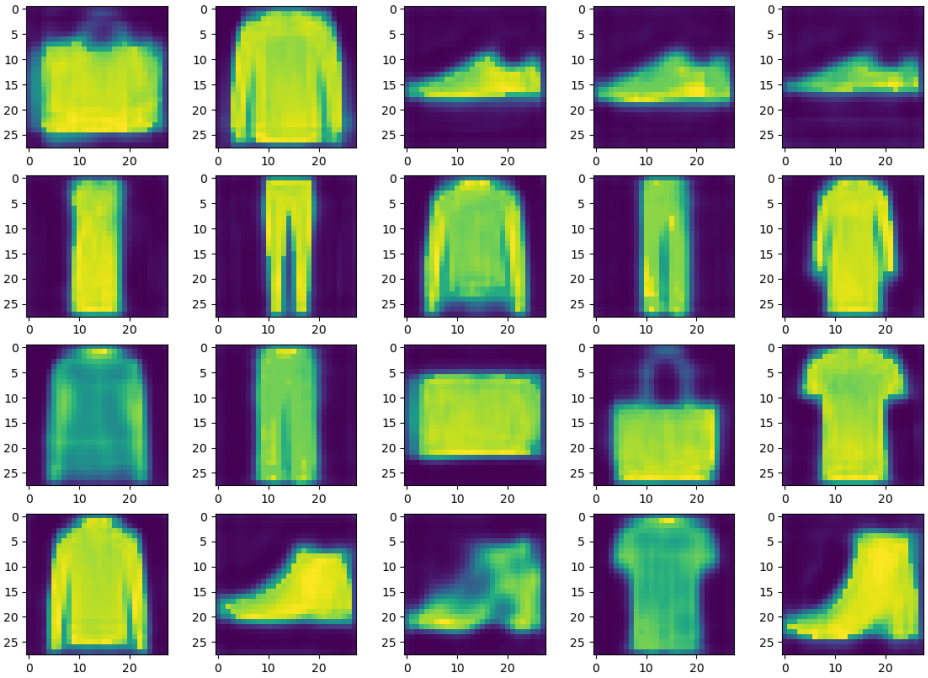

Figure 3.4: Twenty learned pseudo-inputs from a model trained on the Fashion-MNIST dataset, recovered via projection onto the manifold.

changes, the net complexity of the LSV model is roughly equivalent to that of the NVP due to how few added parameters there are. We can see in 3.6 that over the same number of steps (equating roughly to the same amount of time passed as well), the LSV model arrives at a lower loss than the NVP is able to converge to.

Regarding the suitability of the latent space for clustering, we can measure this indirectly by considering the performance of our algorithms on the various spaces. In particular, we evaluate the ARI of a vanilla HDBSCAN algorithm run on the various spaces. We compare this to the two-dimensional UMAP projection as a gold-standard result in table 3.2. This indirect heuristic is fundamentally limited in what it can tell us, since it relies on HDBSCAN's performance. Despite HDBSCAN being free from many of the biased assumptions that beset standard unsupervised clustering algorithms, it still necessarily has its own sets of biases. These include, as mentioned in chapter 2, the notion that clusters are well-separated, which fails in dense representations (something that is improved by PCH).

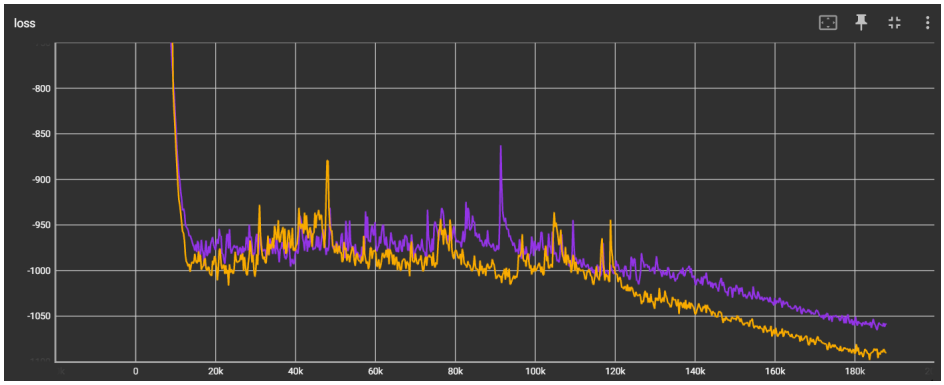


Figure 3.5: The losses of a vanilla VAE (purple) and NVP (orange) model over the course of 400 epochs.

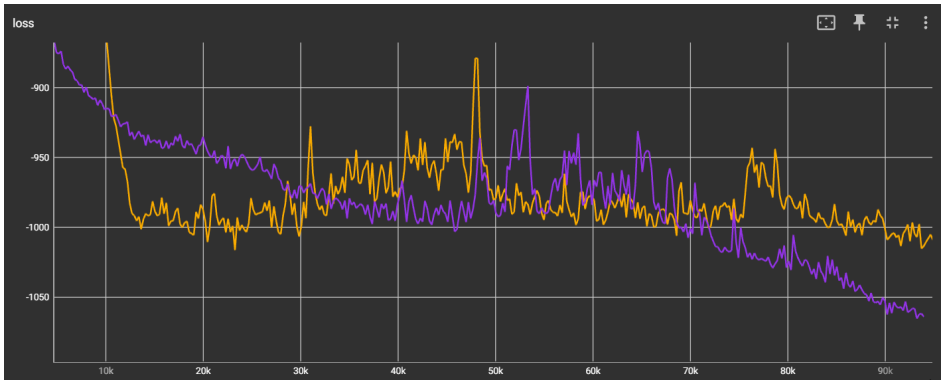


Figure 3.6: The losses of an LSV (purple) and NVP (orange) model over the course of 200 epochs

To demonstrate that the representation power of the methods listed extends past what appears at first glance, we consider a case study on the embedding space of an LSV model. As we can see in its embedding in figure 3.8, the points are densely packed and cluster separation is ambiguous at best. This does not mean we lose the sense of what clusters may exist, but rather that the information is encoded differently. Take, for example, the cover of pseudo-inputs presented in figure 3.9. While these pseudo-inputs do not correspond directly to semantic clusters per-se (though it may be reasonable to take this as a criticism of the ground truth labels rather than the pseudo-inputs) they do still represent local groupings. In fact, if we prune these pseudos intelligently down to what is shown in figure 3.10, we can generate a set of clusters by considering the likelihood of each point under each pseudo-input, and assigning a label to each point corresponding to the pseudo-input with which it has the greatest likelihood. This is pictured in figure 3.12. Such a strategy alone lends an ARI of approximately **0.35**, which is similar to the score we obtain using HDBSCAN. Keep in mind, we did not use any algorithm on top of our model, we simply took a straightforward maximum likelihood estimate approach under the (pruned) pseudo-inputs, and were able to obtain a reasonable clustering.

To emphasize the virtue of the maximum likelihood strategy, and what it means for the representation space, it is worth discussing FMNIST-5 which is a re-labeling of the Fashion-MNIST dataset that reduces the labels to only five distinct ones broken down as {Tshirt/Top, Dress}, {Trouser}, {Pullover, Coat, Shirt}, {Bag}, {Sandal, Sneaker, Ankle Boot} [57]. This is done to respect the ambiguities and overlap within the original Fashion-MNIST, in hopes of creating labels more consistent for machine-learning applications. In this case, evaluating under this re-labeling, we obtain an ARI of **0.52**. To be clear, this is competitive with HDBSCAN, PCH, and COP-KMeans while only making use of a *two-dimensional* embedding space. In fact, this score *beats* other widely cited fully-unsupervised algorithms on FMNIST-5, such as ClusterGAN with an ARI of 0.48, agglomerative clustering with an ARI of 0.36, and non-negative matrix factorization with an ARI of 0.40, as seen in table 3.1.

In summary, NVP offers a drop-in solution for augmenting VAEs with an improved VampPrior such that the pseudo-inputs actually learn to resemble genuine data, leading to simple and robust data introspection through focusing on the pseudo-inputs as learned representatives. Furthermore, we demonstrate that NVP has *greater* learning capacity in even low-parameter models (approximately one million). Finally, we introduce LSV, which allows for us to improve the geometry of the embedding space by encouraging greater locality and density due to the stochastic categorical prior, and we demonstrate its ability to

Table 3.1: The ARI of various models applied to the FMNIST-5 algorithm. Note that PCH and COP-KMeans were given twenty constraints. The best semi-supervised, and fully unsupervised scores are in bold.

| Model | ARI |
|---|---|
| PCH | **0.63** |
| COP-Kmeans | 0.48 |
| LSV MLE | **0.52** |
| ClusterGAN | 0.48 |
| AC | 0.36 |
| NMF | 0.40 |

Table 3.2: The ARI of a vanilla HDBSCAN run on the latent space produced by each representation method.

| Model | HDBSCAN ARI |
|---|---|
| VAE | 0.14 |
| NVP | 0.25 |
| LSV | 0.36 |
| UMAP | **0.40** |

generate high-quality embedding spaces. We also demonstrate that the pseudo-inputs under LSV correspond highly to meaningful local clusters and that light pruning can generate consistent clusters that match or outperform other SOTA algorithms for free on top of LSV. Thereby we implement and confirm our theoretical findings, demonstrating their feasibility as tools for data introspection and semantically meaningful representation learning.

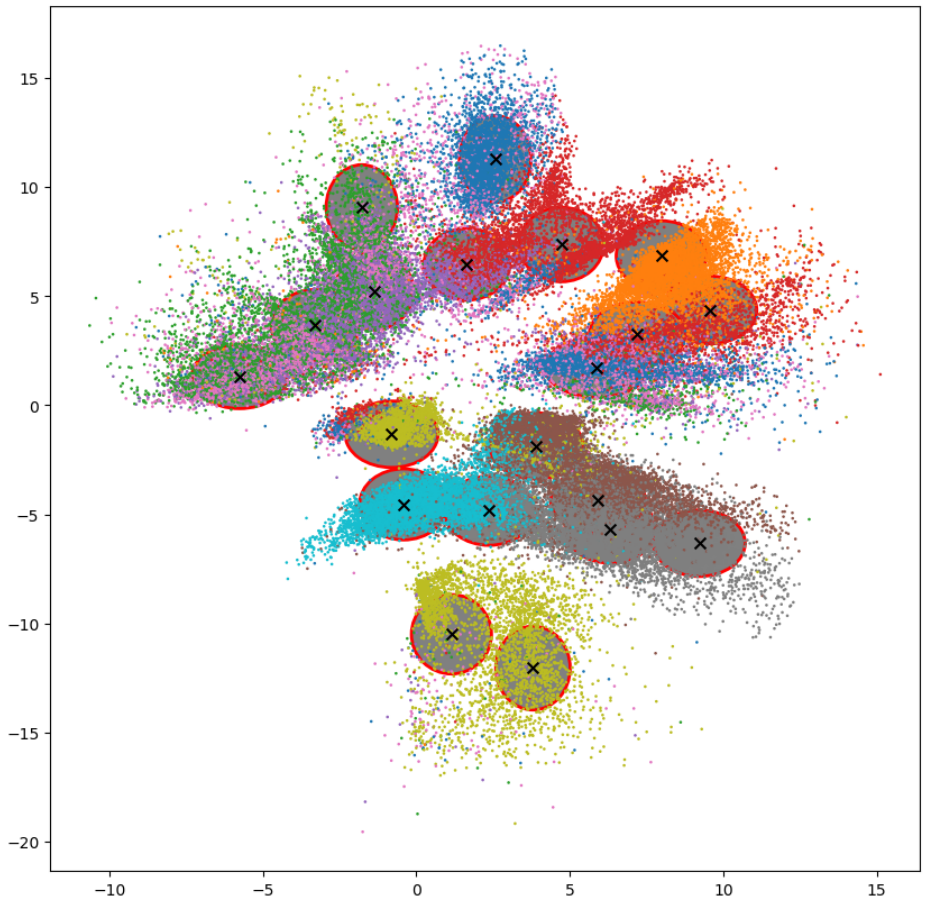


Figure 3.7: The embedding space of a trained NVP model, with pseudo-input posterior distributions marked as ellipses with crosses at the center. The ellipses are drawn to one standard deviation along each axis for the distributions they represent.

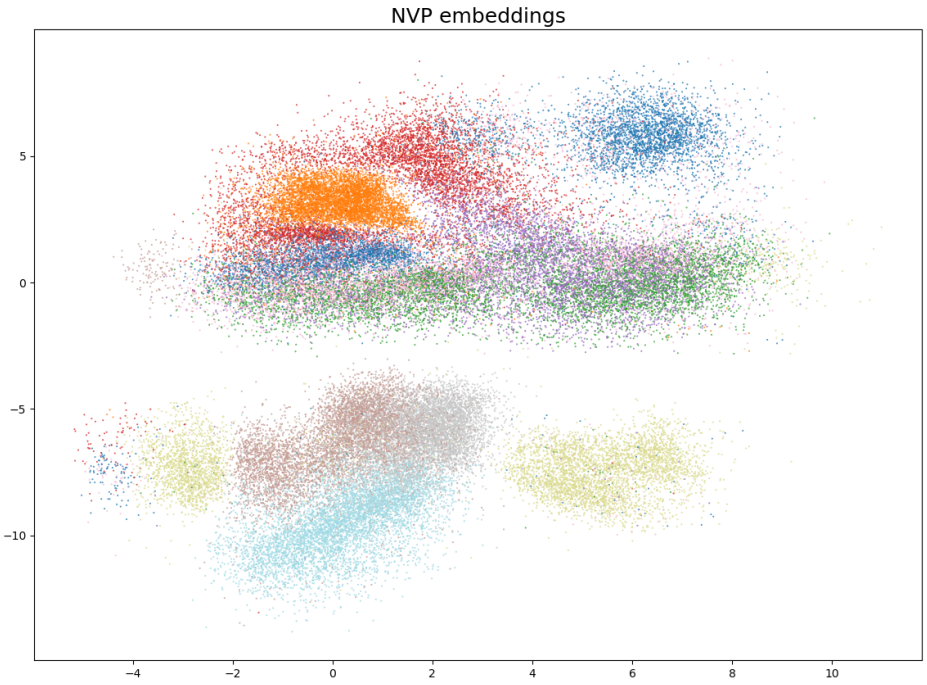


Figure 3.8: The embedding space of a trained LSV model.

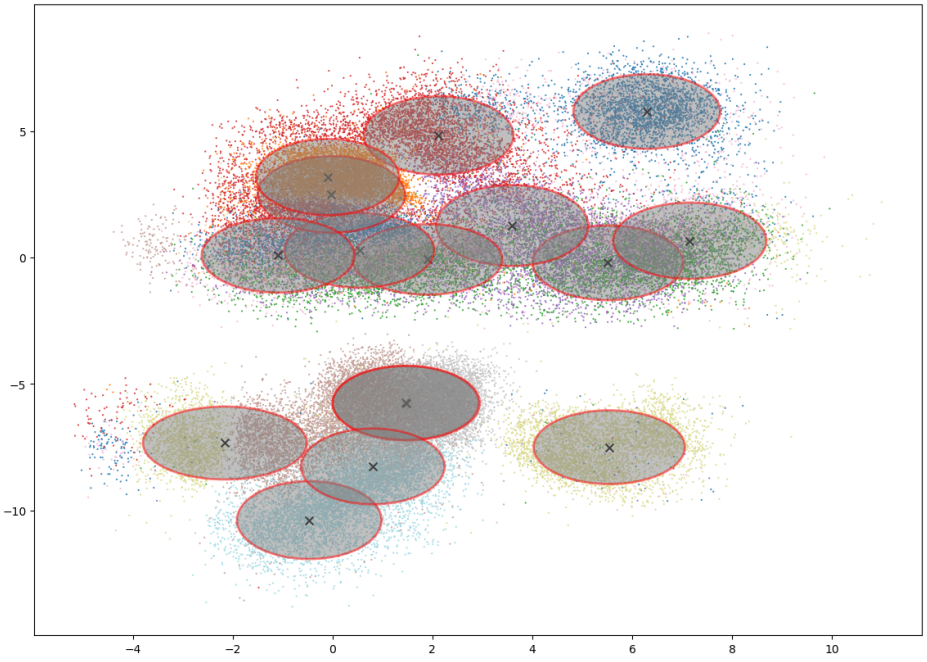


Figure 3.9: The embedding space of a trained LSV model, with pseudo-input posterior distributions marked as ellipses with crosses at the center. The ellipses are drawn to one standard deviation along each axis for the distributions they represent.

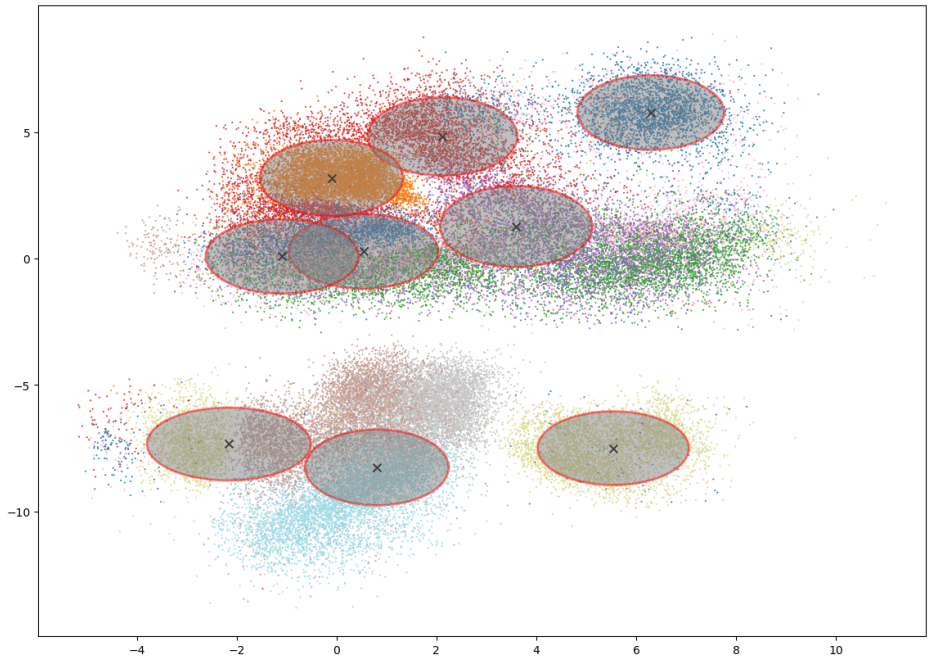


Figure 3.10: The embedding space of a trained LSV model, with a pruned set of pseudo-input posterior distributions.

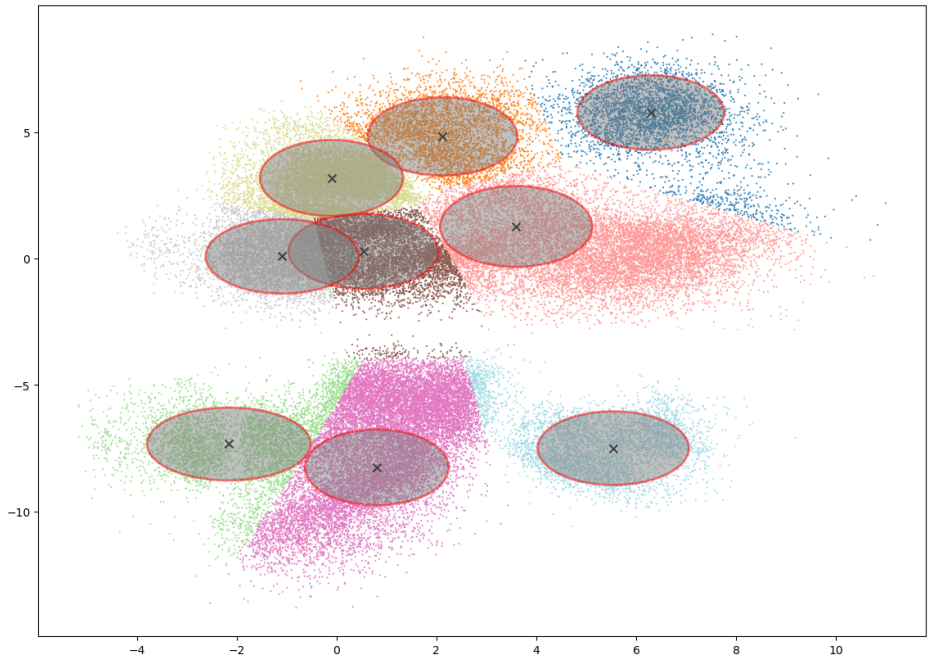


Figure 3.11: The clusters generated by using a maximum likelihood estimation technique with respect to the pseudo-inputs' posterior distributions in an LSV embedding space.

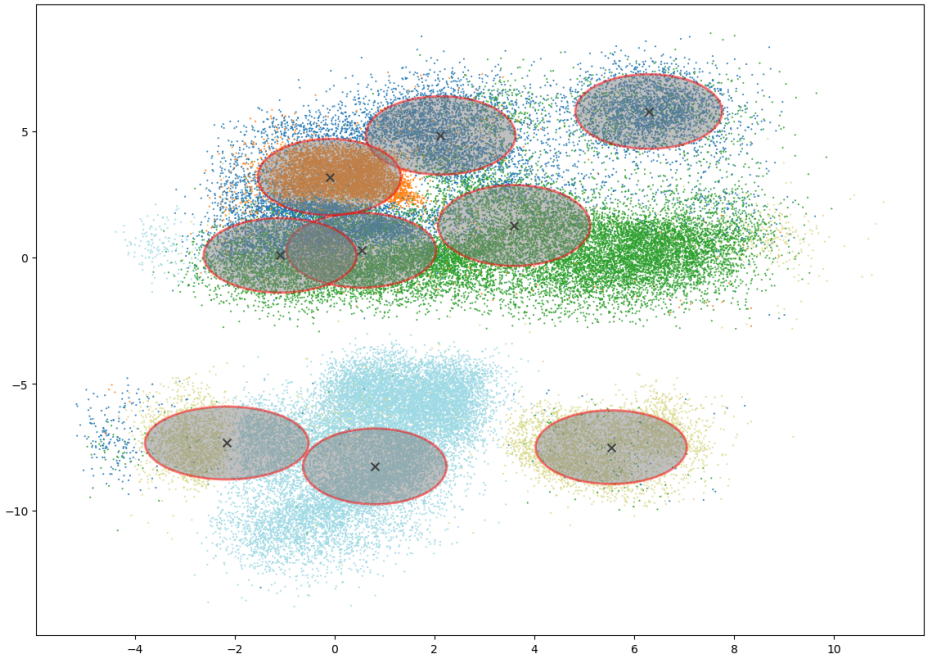


Figure 3.12: The embedding space of a trained LSV model, with a pruned set of pseudo-input posterior distributions. Points are colored by their ground-truth labels under FMNIST-5.

# Chapter 4

# Iterative Refinement

## 4.1 Introduction

While we have introduced methods both to better represent the underlying feature space for our data and to cluster our data using semi-supervision by way of pairwise constraints, we must still develop a third critical aspect of the framework: the infrastructure for iterative refinement. Although plenty of focus goes towards one-pass algorithms, research is not a one-pass task. As we analyze novel data, we gain novel insights, and ideally we can leverage these insights in future analyses to help us learn more accurate and nuanced conclusions from our initially overwhelmingly complex dataset. We can apply a similar workflow to our framework to mitigate key problems that arise in semi-supervised tasks like our own.

For example, despite the surplus of raw, unlabeled data, the availability of labeled or feedback-paired data is greatly hindered by the expensive and time-consuming nature of manual expert labeling. Therefore, purposeful and intentional use of labeled data is essential in many resource-constrained contexts, such as research projects with limited funding, volunteer efforts, non-profits, and other research contexts which face a lack of direct funding.

To minimize barriers to entry and make the most use of the data we already have, we develop a framework which allows for iterative development by two means:

1. intelligently engaging a domain expert to offer feedback on samples that are viewed as informative and important.

2. using the accumulated semi-supervised understanding of the dataset to tweak and refine the algorithmic procedure for the next iteration to encourage better performance.

The first item is generally referred to as "active learning" [72, 79, 48] and the latter item as "self-supervised" learning [27, 37, 59]. A large portion of self-supervised learning is performed in the context of contrastive learning based on self-curated similarity / dissimilarity pairs [13, 59], however the technique itself generalizes to more methods, including the general iterative framework we will later introduce.

In particular, we introduce

1. a mechanism to leverage the pushforward measure introduced in the previous chapter to generalize the capacity of PCH to use traditional statistics and information-theory-based active learning sampling techniques.

2. an iterative HDBSCAN-first active querying method to prioritize highly useful samples for generating pairwise constraints, leveraging the spatial distribution of points and HDBSCAN results directly for partial self-supervision.

In practice, this form of self-supervision is consistent with several representation-based active-learning concepts, thus we will conceptualize it as an active learning technique so as to disambiguate it from the otherwise large field of self-supervision.

## 4.2 Related Works

Active learning attempts to find a set of high-quality data points that are the most "informative" and "representative" points in the dataset with respect to a given model such that labeling those points will have the highest return of similarly-sized data subsets. [79]. Ultimately the study of active learning relates a model and its latent information with the underlying dataset so as to recommend queries for feedback from experts – also called *oracles*. These queries can range from direct and explicit labels to indirect or implicit comparisons (e.g. "triplet loss" as used in metric learning) [72]. In their landmark active learning survey, Settles divides active learning scenarios into three major components [72]:

1. Membership query synthesis, whereby the machine learning system or algorithm itself generates a new data point which may be particularly informative (e.g. one that lies on a decision boundary) which is then queried for feedback, directly allowing the model to choose the information it receives through manipulation of the query synthesis.

2. Stream-based selective sampling, whereby the model is faced with an ongoing stream of potential query samples, deciding directly on which to query or discard.

3. Pool-based sampling, whereby the model has a large pool of unannotated data a priori and must select samples from within that pool to query.

Note that the first scenario can be rather difficult to coordinate when the oracle is a human, since many potential membership queries can – in the words of the manifold hypothesis – lie on the boundary of the manifold or off-manifold entirely, resulting in the synthetic data point resembling semantically ambiguous or even meaningless data. This is less of an issue when the data manifold is more aligned with the ambient space, e.g. in the case of a phase space for robotics programming, where each point in the ambient space can correspond to a sensible configuration for the robot to occupy, hence synthesized queries pose little challenge. The opposite case would be e.g. semantic data such as images, text, speech, etc. where the data manifold is a complex structure, and perturbations in the ambient space align more with "corruption" than smooth deformation. In general, more complex systems are poorly suited for membership query synthesis, especially with human oracles that are ill-suited to off-manifold semantic data [99].

Stream-based queries take as premise the notion that obtaining unlabeled instances of data is relatively free and can be done on demand (which is a valid assumption in many modern cases, e.g. web-scraped datasets). The model learns by repeatedly being prompted with a new data point, then deciding whether to discard the point or to query the oracle regarding it. Granted that the distribution of data is fixed, this method can be understood as a means of generating the direct membership query scenario since each point will *eventually* be seen with probability one. However, this scenario is most useful for describing situations where the incoming data has some genuine novelty and is unlikely to be recovered later, in which case the immediate decision on whether to query the data has a sense of immediacy to it. In the case of an unknown distribution, due to the underlying assumption of stochasticity in the sampling of the data, we can be assured it will still follow the ground-truth distribution despite our lack of a tractable representation of the distribution itself.

Finally, pool-based sampling refers to the situation where we have a labeled subset of otherwise-unlabeled data. These pools are usually assumed to be fixed, distinguishing pool-based sampling from the stream-based scenario, which directly assumes new samples can be obtained, and the synthesis sampling scenario, wherein new samples are *created*). The fixedness assumption is not strictly necessary, but it simplifies the problem and allows for the development of greedy

algorithms for optimization since we have access to all potential queries a priori. The pool-based sampling method is especially popular in machine-learning contexts and has been used for text classification [53, 47, 13, 33, 84], image processing [100, 83], video classification [92, 28], speech recognition [87], and much more.

The most directly applicable active learning scenario to PCH is the pool-based labeling scheme, since our data is fixed a priori and finite, thus we cannot match the assumptions for stream-based learning. However, the representation learning model from Chapter 3 has the capacity for engaging in membership query synthesis through the use of NVP – since the pseudo-inputs can be learned to lie on the data manifold in the ambient space, they will be semantically similar to actual data, meaning that a human oracle may interpret the pseudo-inputs in the problem context. We discuss this in more detail in the methodology section.

Perhaps the simplest and most straightforward active learning query technique is *uncertainty sampling* [46], where queries are determined by how uncertain the model is regarding their label, i.e. to find

$$x^* = \underset{x}{\operatorname{argmax}} \left(1 - P_\theta(\hat{y}|x)\right)$$

where $\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$. This method has several shortcomings, most prominently in multi-class settings, where most information is, in some sense, "omitted". This omission arises from the absolute determination using only the greatest probability, hence being invariant to the rest of the distribution. In response, several variants have arisen to address these shortcomings, with the most prolific being the use of entropy in place of the aforementioned naïve uncertainty. The entropy-based determination is as follows:

$$x^* = \underset{x}{\operatorname{argmax}}\, \mathbb{H}[P(y|x)] = \underset{x}{\operatorname{argmax}} - \sum_i P(y_i|x) \ln(P(y_i|x))$$

[73] which clearly incorporates more of the underlying information encoded in the *distribution* of probabilities per sample, rather than simple uncertainty. While maximal samples optimize both forms of uncertainty sampling, the distribution of "uncertainty score" differs wildly for non-maximal points. This difference in distribution results in a more robust sampling when using the entropy criterion.

While these sampling strategies are especially well-suited for probabilistic models, there has been success in translating them into deterministic problem contexts using a stochastic ensemble approach to generate a posterior distribution, essentially taking a monte-carlo estimate of a true underlying posterior

[46]. That is to say, given $k$ voting deterministic classifiers (e.g. a random forest), we can define the posterior distribution

$$p_\theta(y|x) = \frac{1}{k}\sum_{i=1}^{k} p_\theta^i(y|x)$$

where $p_\theta^i(y|x)$ represents the "stochastic" posterior of the $i$-th classifier. Generally $p_\theta^i(y|x) = \delta_{\hat{y}_i}(y)$ where $\delta$ is the delta function and $\hat{y}_i$ is the $i$-th classifier's deterministic output label. This simply means that $p_\theta^i(y|x) = 1$ iff $y = \hat{y}_i$. In this way, the aggregate posterior $p_\theta(y|x)$ approximates a hidden posterior $p_\theta^*(y|x)$, which is the ensemble's estimate of the true posterior $p_\theta(y|x)$ [25, 50].

In addition to the information-based approaches presented thus far, there are representation-based query strategies, which generally attempt to leverage the structure of the unlabeled data to find points that best represent the given dataset. The idea is that the information gained from the points' queries will provide meaningful gain to all points in their vicinities, resulting in significant total uplift in performance [79]. In a sense, this strategy is the "opposite" of the information-based approach. Whereas the information-based approach attempts to query boundary points or points of great uncertainty, low information density, and high entropy, representation-style approaches instead attempt to find points well-situated within the data distribution – often centroids or medoids of their local distributions.

Consequently density-based approaches often implement representation based queries through attempting to find cover sets of the data distribution so that all data lie within a certain range to the closest queried representative [89, 71]. Similarly, some methods attempt to cluster the data and select representative points based on those clusters [36, 39]. Ultimately, representation-based query strategies offer a better "spread" over the data, addressing both the tendency of information-based queries to remain dense in the representation space (e.g. attracted by a central region of high uncertainty) and their implicit redundancy (one query may clarify the region, no need for multiple) by explicitly focusing on covering the dataset and matching the implicit geometric structure. In exchange, representation-based methods suffer from their own form of redundancy, wherein points sampled with respect to data density may not be as helpful as information-based queries, as the large concentration of data in that region implies that the machine learning algorithm will be better suited for that region simply due to the distribution of data itself.

## 4.3 Methodology

### 4.3.1 Pushforward for Statistical Measure

While uncertainty-based sampling is a powerful tool in active learning, HDBSCAN by itself (and consequently PCH) is not truly probabilistic and thus is unable to directly apply uncertainty-based query methods. In particular, given a fixed dataset, HDBSCAN is deterministic with a notion of "membership strength" which is a normalized heuristic, but not a true probability in the sense that it does not refer to a stochastic quantity or process.

The aforementioned strategy of amending the estimated posterior distribution of labels based on ensemble classification is theoretically possible in one of two modes: either multiple HDBSCAN models are run with different hyperparameters to build an ensemble or multiple *neighbors* are taken as a source of label votes such that the ensemble is a local neighborhood around each point. The former case is infeasible because the computational costs of running HDBSCAN multiple times are high and the hyperparameters of HDBSCAN are many, and it is difficult to predict how hyperparameter changes will affect the outcome of the algorithm over a given dataset. Furthermore, it is not obvious what weight ought to be afforded to each voting member of the ensemble, and a uniform weighting is unlikely to do well due to the model's nonlinear sensitivity to hyperparameters. On the other hand, an ensemble of neighbors is reasonable for boundary points, where local neighborhoods lead to non-trivial posteriors due to the presence of multiple labels. However, interior points become trivially confident and do not reflect actual uncertainty, e.g. what is an interior point with $0$ entropy now may become a boundary point with high entropy with a small change to either the MST or an arbitrary hyperparameter.

Instead, representation-based methods are usable, yet highly redundant due to the underlying structure of HDBSCAN, wherein labels are assigned on a cluster-representation basis, meaning that HDBSCAN implicitly applies a trivial cluster-based active learning where each query forms a new label. Similarly, due to the algorithm's density-based divisive structure, the same can be said for implicit use of trivial density-based active learning, where differences in local density imply different labels.

Thus to augment HDBSCAN in a unique way, we consider the problem of finding a way to apply uncertainty-based query strategies on this deterministic algorithm. The simplest way to do so would be to generate a sense of distribution over the underlying dataset. This can be completed simply by fitting a $k$-component Gaussian Mixture Model (GMM) over the data, where $k$ is the

number of unique non-noise labels present in the output of HDBSCAN. Once this distribution is established, we can use uncertainty-based query strategies by setting the likelihood of each point to its likelihood under the fitted distribution. Unfortunately, a GMM-based approach relies on the assumption that the data are fundamentally normally distributed within clusters, and that clusters are elliptical – neither assumption is necessarily true in practice and depend greatly on the choice of representation of the data (e.g. t-SNE vs UMAP will give different distributions). One may instead opt for non-parametric kernel-density estimates, but the choice of kernel carries with it assumptions about the underlying distribution of the data in the representation space, which can significantly impact the final distribution and (e.g. spherical/elliptical in the case of most kernels).

How do we generate a distribution over the data without assuming the underlying distribution of the data within clusters and within the space? The two-stage VAE implemented earlier is a rather fitting solution: recall that the key finding from the two-stage VAE discussion was that *even with a Gaussian prior*, a two-stage VAE system can globally optimize the VAE objective while recovering both the geometric manifold of the data and a sense of measure (i.e. distribution) over the data manifold. Specifically, we recall the pushforward measure defined in section 3.4.2, which allows us to use the learned two-stage VAE to define a distribution over the ambient space, under which we may calculate the likelihood of the data. In particular, for $x \in \mathcal{X} \subset \mathbb{R}^n$ with pushforward measure $\lambda_p = \lambda_d \circ g \circ f$[12], we can derive likelihood by considering the Radon-Nikodym derivative of the induced pushforward measure with respect to the manifold-dimension Lebesgue measure

$$\frac{d\lambda_p}{d\lambda} = \frac{d\lambda_d}{d\lambda} \circ g \circ f,$$

which can be rewritten into

$$p_\theta(x) = p_\theta^u \circ g \circ f$$

where $p_\theta^u = \frac{d\lambda_d}{d\lambda}$, corresponding to the prior of the second-stage VAE which is generally assumed to be $p_\theta^u(x) = \mathcal{N}(0, I_d)$. Now we have defined a concrete, tractable $p_\theta(x)$ that assigns likelihood to the data points, enabling uncertainty-based calculations that rely on the geometric and spatial distribution of points but not the label distribution.

[12] Note that here we use $f, g$ in their capacity as *maps* rather than outright functions, mainly to transform images across domains, hence why we do not apply what may seem like the "usual" calculus in later applying the Radon-Nikodym derivative.

### 4.3.2 Radial Constraint Sampling

In addition to our use of uncertainty-based active learning queries through the pushforward measure, we also introduce a novel pairwise-constraint query method called "Radial Constraint Sampling" (RCS). The key idea of RCS is to offer an active learning query method that directly leverages the assumptions of HDBSCAN in a way that will hopefully augment PCH most directly. We note that HDBSCAN operates off of an MRD MST, so each edge in the MST corresponds to a "close neighbor" with respect to MRD [13] and thus borrows the ambient space's metric for determining connectivity. While uncertainty-based methods focus on statistical properties, they are often plagued with problems of poor spatial distribution with respect to spatial diversity and uniformity. Meanwhile, representation-based and density-based query strategies often overfit the spatial heuristics they optimize for, resulting in an uninformed uniform distribution of queries according to the particular biases of the query method.

[13] Note that the spanning nature of the MST requires that they are not strictly the closest points, but they will often be within $k$ neighbors thus it suffices to consider them sufficiently local. The number of exceptions are generally few, and correspond to the number of clusters.

Our proposed method of RCS addresses aspects of both schools of thought; it favors sampling highly-informative points based not on statistical quantities, but algorithmic properties of HDBSCAN itself. It optimizes for geometric heuristics based on the clustering bias of PCH which is comprised of *dividing* clusters through CLCs, and *combining* clusters through MLCs.

Formally, we perform RCS by uniformly sampling constraints between inter-cluster and intra-cluster pairs of points. For inter-cluster pairs, we first start by sampling *anchor* points from a cluster $C$ based on the square of their pairwise ambient-space distance to their closest same-cluster neighbor, written as

$$w_a(x_i) = \min_{x' \in C \setminus \{x_i\}} \|x_i - x'\|^2$$

For each anchor point, we then sample a *bridge* point outside of the cluster, weighing the probability each point is chosen based on their inverse-square distance to the anchor point

$$w_b(x_i, y_j) = \|x_i - y_j\|^{-2}.$$

This method generally samples anchor points on the boundary of a cluster (where their nearest-neighbor distance is large) and pairs them with bridge points on the closest external boundary, acting as a means to clarify whether two adjacent clusters ought to be considered the same or distinct, based on the farthest pairwise-compatible outliers of each (see algorithm 4).

For intra-cluster pairs, we begin by sampling an anchor point from the cluster with points weighted by the square of the distance to their farthest neighbor

$w'_a(x_i) = \max_{x' \in C \setminus \{x_i\}} \|x_i - x'\|^2$. We then sample a bridge point within the cluster $C$, weighing the probability each point is chosen by the square of their distance to the anchor point,

$$w'_b(x_i, y_j) = \|x_i - y_j\|^2.$$

With these choices of weights, inter-/intra-cluster sampling act as duals. Whereas our inter-cluster sampling method seeks the *closest* pair of relative outliers, intra-cluster sampling wants to find the *farthest* pair of relative outliers. The inter-cluster sampling asks "should we merge these clusters, or keep them separate" while intra-cluster sampling asks "should we fragment this cluster, or keep it whole", offering two distinct and exclusive ways to generate insightful MLCs/CLCS.

Of course, this method requires knowing clusters a priori, which would render it moot for a single-pass constrained clustering algorithm. However, this is easily solved by utilizing the outputs from a first pass of an unsupervised HDBSCAN. We refer to this as "estimate-based RCS", which is the most readily available tractable mechanism for RCS in the given problem context. Interestingly, this enables the prospect of iteratively performing estimate-based RCS, updating the cluster identities every run. This kind of iterative estimate-then-query approach is closely aligned with the classical approach to data investigation, wherein with each new insight gained, the bias regarding *where to explore next* gets updated.

### 4.3.3 Pseudo-Input Analysis

Finally, to streamline the analysis of novel datasets using the currently established framework of PCH in addition to a two-stage NVP model, we discuss the nature of pseudo-inputs, generative distributions, and the insights and decisions that come with them. This section focuses on human intervention and allowing expert semantics to interfere directly in the representation of the model in a way that affects all other tasks.

First, we discuss *prototype discovery*, a useful ability of the two-stage NVP model. When learning a proper representation, the two-stage NVP model optimizes for an aggregate posterior which acts as an estimate to the aggregate prior distribution,

$$\underset{q}{\operatorname{argmax}}\, \mathbb{E}_{z \sim p(z)}[q(z)] = \frac{1}{N} \sum_{i=1}^{N} q_\theta(z|x_i) \approx \frac{1}{K} \sum_{i=1}^{K} q_\theta(z|x_i^\eta)$$

**Algorithm 4** Inter-cluster Radial Sampling

**Require:** $K$, the number of constraints to sample. $\{x_i\}_i^N = \mathcal{X} \subset \mathbb{R}^n$, a set of points from which to sample. $\{y_i\}_i^N = L$, a set of labels corresponding to an estimated labeling $(x_i, y_i)$, and a label-assignment function $l(x_i) \rightarrow y_i$. An oracle/expert to query, with a corresponding ground-truth similarity function $s$ such that $s(x_i, x_j) = 1$ iff the two points have the same ground-truth label. Note that this does not require *knowing* the label, but rather knowing that they share one.

**Ensure:** A list of MLCs $M$ and a list of CLCs $C$ such that $|M| + |C| = K$, where constraints are set as MLCs/CLCs based on the estimated labeling of their comprising points.

$M \leftarrow \{\}$
$C \leftarrow \{\}$
$w_i \leftarrow 0$ for $i \in \{1, \ldots, N\}$
$W \leftarrow \{w_1, \ldots, w_N\}$
**for** $y$ in $\text{Unique}(L)$ **do**
  $S \leftarrow \{x_k \in \mathcal{X} \mid l(x_k) = y\}$
  **for** $x_i$ in S **do**
    $w_i \leftarrow \min_{x_j \in S} \|x_i - x_j\|^2$
  **end for**
**end for**
**for** $i$ in $\{1, \ldots, K\}$ **do**:
  $u_i \leftarrow \text{Sample}(\mathcal{X}, W)$
  $S \leftarrow \{x_k \in \mathcal{X} \mid l(x_k) \neq l(u_i)\}$
  **for** $x_j$ in $S$ **do**
    $w'_{i,j} \leftarrow \|u_i - x_j\|^{-2}$
    $W'_i \leftarrow \{w'_{i,1}, \ldots, w'_{i,N}\}$
  **end for**
  $x_j \leftarrow \text{Sample}(S, W'_i)$
  **if** $s(u_i, x_j) = 1$ **then**
    $M \leftarrow M \cup \{(u_i, x_j)\}$
  **else**
    $C \leftarrow C \cup \{(u_i, x_j)\}$
  **end if**
**end for**
  **return** $M, C$

which not only learns the general aggregate prior distribution $\frac{1}{K}\sum_{i=1}^{K} q_\theta(z|x_i^\eta)$, but importantly also the pseudo-inputs $\{x_i^\eta\}_{i=1}^k$ which parameterize the distribution. As established by Zain et al., when these pseudo-inputs are learned in accordance to NVP-style training (the additional pseudo-input loss regularization term), the learned pseudo-inputs are qualitatively similar to true data samples, yet may exhibit minor artifacts due to being partially off-manifold[14] [98]. Furthermore, the use of NVP-style loss and training does not decrease the efficacy of the model itself when compared to the usual VampPrior implementation [98].

[14] This will always be an issue since a true data manifold will almost always have zero volume in the ambient space, and hence the best we can do is to learn the noisy manifold estimate, meaning some points will be off-manifold despite seeming normal.

This means that they are uniquely well-suited to introspection, and can play the role of "prototypes" for the data distribution as a whole. These learned prototypes can then be inspected directly, with their structure offering insights to the domain expert regarding the structure and distribution of the underlying data. Take, for example, figure 3.4, which shows how the set of pseudo-inputs can serve as a learned "representative" sample of the underlying data. Note that the pseudo-inputs will almost surely have several redundancies within them as part of their design – the number of pseudo-inputs ought to be significantly greater than the number of suspected classes to increase the chance of obtaining a representative from each interesting class. Pruning pseudo-inputs is relatively straightforward as well, meaning that you can simplify the model and remove pseudo-inputs which are deemed unnecessary or redundant. Note that redundancies occur in proportion to the relative density of classes within the dataset as a whole, and therefore contain information regarding the data distribution. It is not necessarily "optimal" (with respect to model loss) to prune these redundancies, however it can easily be done in the case a one-to-one relationship between pseudo-inputs and classes is desired.

Manual inspection for redundancies is viable in the case of relatively simple data and models with relatively few pseudo-inputs, but quickly becomes intractable for models with a great number of pseudo-inputs. Therefore, we determine the relative redundancy of pairs of psuedo-inputs by considering their relative distance. A naïve approach would consider the distance between two pseudo-inputs with respect to the ambient data metric, but distances in the ambient space correspond poorly to semantic factors due to the dimensionality gap with the underlying data manifold, and consequently measures noise more than semantic dissimilarity. A more informed approach is to consider not the pseudo-inputs directly, but the posterior distributions they induce, i.e. $q_\theta(z|x_i^\eta)$, in which case we need to utilize a statistical distance. A classic go-to statistical distance heuristic is KL-divergence, which fails to be a proper distance due to its asymmetry in its arguments. Instead, we opt for the Hellinger dis-

tance [30]. Specifically, we define the squared Hellinger distance between two distributions $P, Q$ as

$$H^2(P,Q) = \frac{1}{2}\int_{\mathcal{X}} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \lambda(dx)$$

where we take $\lambda$ to be an auxiliary measure such that both $P, Q$ are absolutely continuous with respect to $\lambda$. Note that the existence of such a measure is guaranteed and can be trivially taken as $\lambda = P + Q$. Then, the functions $p, q$ are defined as the Radon-Nikodym derivatives $\frac{dP}{d\lambda}, \frac{dQ}{d\lambda}$ respectively. Note that the choice of $\lambda$ does not affect the final value of the Hellinger distance, only its computation (via determination of $p, q$). In particular, taking $\lambda$ to be the standard Lebesgue measure, we can rewrite

$$H^2(P,Q) = 1 - \int_{\mathcal{X}} \sqrt{p(x)q(x)}dx$$

When both $P, Q$ are parameterized Gaussian distributions $P = \mathcal{N}(\mu_1, \Sigma_1),\ Q = \mathcal{N}(\mu_2, \Sigma_2)$ we have that

$$H^2(P,Q) = 1 - \frac{|\Sigma_1|^{\frac{1}{4}}|\Sigma_2|^{\frac{1}{4}}}{|\frac{1}{2}\Sigma_1 + \frac{1}{2}\Sigma_2|^{\frac{1}{2}}} \exp(-\frac{1}{8}(\mu_1-\mu_2)^\intercal(\frac{1}{2}\Sigma_1+\frac{1}{2}\Sigma_2)^{-1}(\mu_1-\mu_2))$$

Alternatively, one may also consider the Jensen-Shannon divergence (JS-divergence) [49] which is a symmetrization of the standard KL-divergence for distributions $P, Q$:

$$D_{JS}(P,Q) = \frac{1}{2}\left(D_{KL}(P,Q) + D_{KL}(Q,P)\right)$$

Note that Jensen-Shannon divergence is not directly a distance metric, but its square-root is:

$$d_{JS}(P,Q) := \sqrt{D_{JS}(P,Q)}$$

Both distance measures offer alternative mechanisms for comparing the similarity of two distributions, and preference between them is at this point merely heuristic.

Thus pruning pseudo-inputs may be as simple as a matter of removing pseudo-inputs that have the lowest pairwise distance from their posterior distributions to other pseudo-inputs' posteriors. An important nuance, however, is to consider *which* pseudo-input to prune when considering a pair with low pairwise distance. Pruning both would be erroneous, since they are only necessarily redundant with respect to *each other*, so we must choose one of the

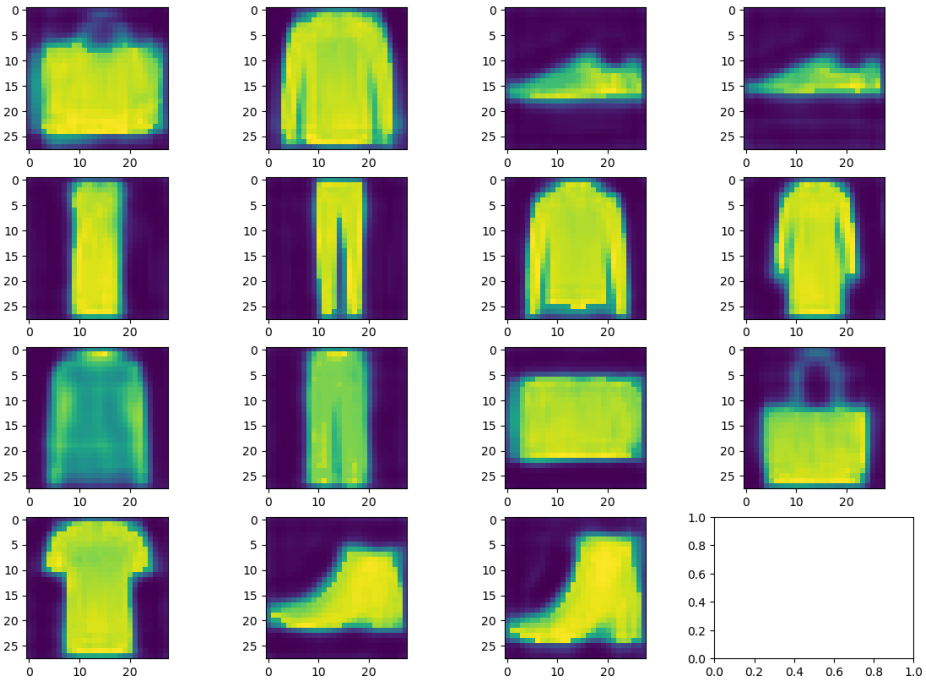

Figure 4.1: Fifteen learned pseudo-inputs from a model trained on the Fashion-MNIST dataset, recovered via projection onto the manifold, after having redundancies pruned.

two. To do this, we consider again the KL-divergence and specifically prune the distribution with the lowest directed KL-divergence to its partner, i.e.

$$\underset{P}{\operatorname{argmin}}\, D_{KL}(P, Q)$$

where with some abuse of notation we take $Q$ to be the "other" distribution for a given $P$. Consequently, we remove the pseudo-input with the lower directed KL-divergence and proceed as normal.

For example, we can prune five pseudo-inputs in this way from figure 3.4 to arrive at a total of fifteen pseudo-inputs in figure 4.1. While the pruned version seems to still cover the dataset will and represent all the data, there still seems to be some room for culling redundancies. At this point, we must consider a mechanism for suggesting *how many redundancies there may be* in a given set of learned pseudo-inputs. Utilizing the notion of sample-entropy and batch-entropy mentioned in section 3.4.3 we can consider two heuristics: the entropy

gap $\alpha$, and the cumulative batch entropy

$$\mathbb{H}_j[\mathbb{E}_i[p_{ji}]] = \frac{1}{N}\left(\sum_{i=1}^{N}\sum_{j=1}^{K} p_{i,j}\ln(p_{i,j})\right)$$

. In practice, the cumulative batch entropy serves as a better heuristic due to the fact that the entropy gap $\alpha$ is contingent on the ability of the network to specify exactly which posterior a data point ought to be attributed to, which will never be perfect since in practice the semantic overlap between posteriors implies that there ought to be a stochastic chance that a sample originates from one of *many* posteriors.

Interestingly enough, it is not necessary for the pseudo-inputs to greatly resemble data for them to be useful. Indeed, *poor quality* pseudo-inputs convey the fact that they may represent regions with greater uncertainty (leading to reduced reconstruction quality) which may itself be valuable for directing manual inspection and labeling. In a way, this is a *subjective* uncertainty measure, similar to quantitative "subjective" information scores such as the Fréchet inception distance (FID) [31]. In a way, the pseudo-inputs reflect the subjective and qualitative understanding of the model regarding how to represent the data distribution.

Ultimately, the use of NVP and the local two-stage VAE framework enables a great deal of augmented data introspection in a semi-automated way. Each step of introspection focuses on allowing a domain expert to make the ultimate determination, yet affords them guidelines and suggestions in both explicit recommendations (e.g. pseudo-input redundancy scores) and open-ended options (e.g. introspection of pseudo-input quality). The feedback mechanisms exist both within the framework itself (e.g. pseudo-input pruning) and outside the framework, in the realm of data curation (e.g. augmenting data corresponding to blurry/uncertain pseudo-inputs).

## 4.4 Results

### 4.4.1 Radial Constraint Sampling

HDBSCAN in a semi-supervised context requires careful consideration of how to employ active learning and partial information. In fact, naive methods can actually *decrease* the performance of both the original baseline method for HDBSCAN, as well as PCH. We evaluate the proposed radial sampling method over both PCH and the baseline method and show the relative performance

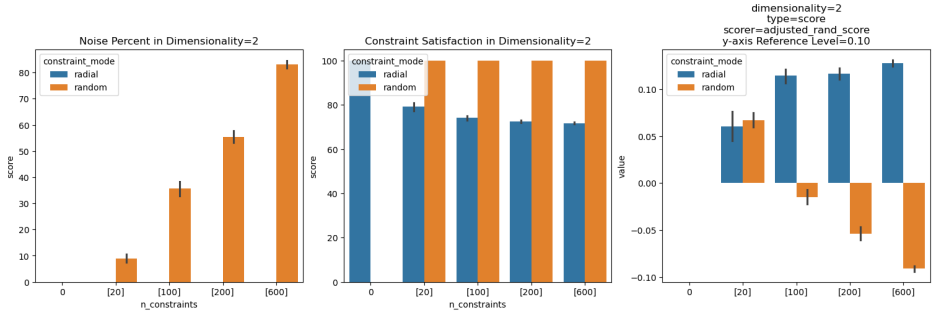


Figure 4.2: A comparison of the relative efficacy of radial and uniform sampling over the Wine dataset when using the original baseline algorithm for semi-supervised HDBSCAN.

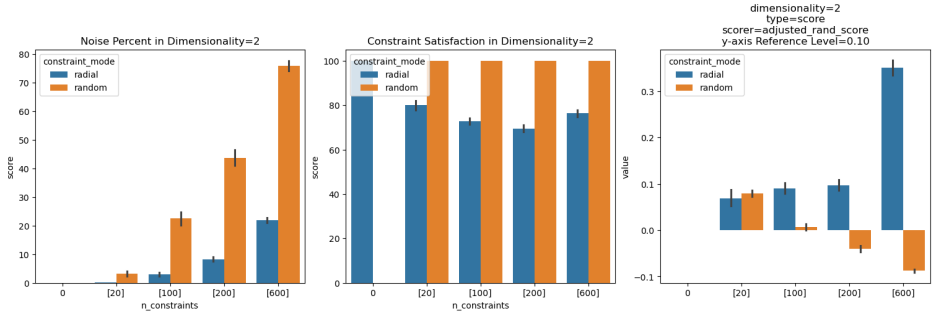


Figure 4.3: A comparison of the relative efficacy of radial and uniform sampling over the Wine dataset when using PCH.

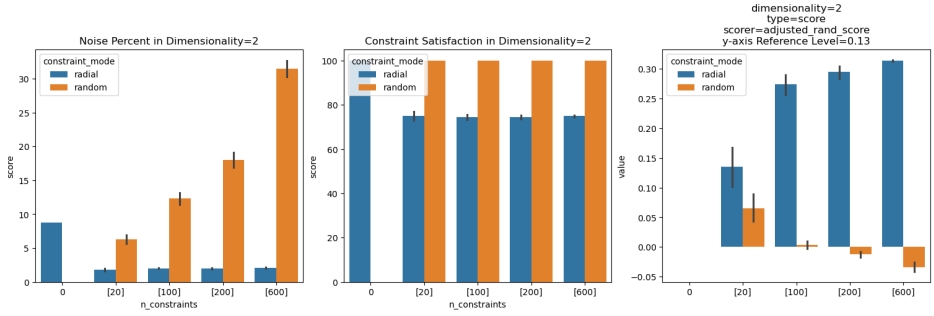


Figure 4.4: A comparison of the relative efficacy of radial and uniform sampling over the Anuran Calls dataset using species-level labels when using the original baseline algorithm for semi-supervised HDBSCAN.

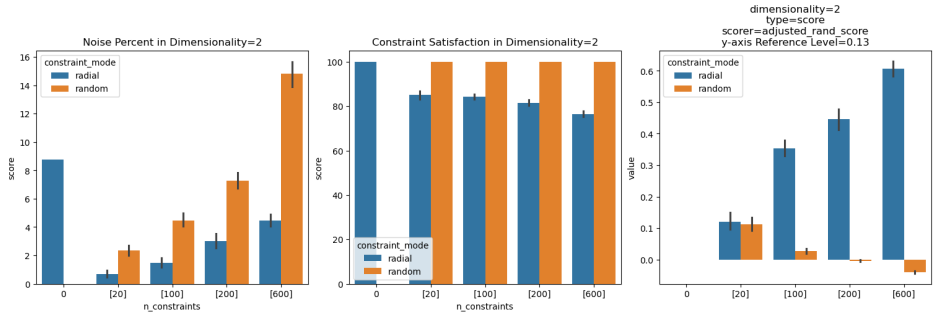


Figure 4.5: A comparison of the relative efficacy of radial and uniform sampling over the Anuran Calls dataset using species-level labels when using PCH.

gains/losses as compared to a control run of vanilla HDBSCAN, similar to the benchmarks in chapter 2, which we refer the reader to for the sake of greater in-depth comparison.

We begin by noting that, as previously mentioned, a poor approach to including partial information can actually *harm* the models' performances. We demonstrate this by utilizing a uniform sampling pairwise constraints, wherein both points within a constraint are sampled randomly, and the nature of the constraint as an MLC or CLC is decided by the ground-truth labels, such that it mimics the response offered by an expert. We can see the relative performances of our proposed radial method versus a naïve uniform method in figures 4.2, 4.3 which evalute both the original baseline method and PCH over the Wine dataset, and figures 4.4, 4.5 which do the same over the Anuran Calls dataset with species-level labeling. We can see in these cases that uniform sampling actually tends to make performance *worse*.

Specifically, we can infer that the sampling methods induce extreme amounts of noise in the algorithms' outputs, which result in a lowered ARI due to the inability to cluster many, or even the *majority* of points. Radial sampling does not have this issue, and indeed shows strong performance boosts for both the baseline method, and PCH with the latter gaining by far the most uplift, most likely due to having a higher representation capacity so as to make use of the information afforded by the radial method. We note that while constraint satisfaction seems to decrease for both methods when using PCH, it is generally due to the algorithms being able to actually find new clusterings that can consistently satisfy the radially-sampled constraints, which may sometimes be at the expense of a few constraints, whereas with uniform sampling the clusterings are near-identical or even worse compared to the fully unsupervised clusterings.

### 4.4.2 Pseudo-Input Analysis

Having access to pseudo-inputs which are able to learn to mimic/emulate genuine data allows a brand new type of analysis, which we term surrogate analysis. We consider the pseudo-inputs as representative, approximately-on-manifold samples from an estimate of the data distribution generated by the underlying VAE model. This means that they jointly encode both information from the underlying data, and from the underlying VAE model, which can be trivially observed from their construction as learned parameters. We proceed with pseudo-input based surrogate analysis case study, abiding by the following procedure over the Fashion-MNIST dataset:

1. Initial observations and tuning of expert sentiment.
2. Estimates regarding the number of geometry-driven clusters.
3. Calculation of relative statistical distances between pseudo-input posteriors.
4. Ranking of pseudo-input pairs based on statistical distances for pseudo-input pruning proposals.
5. Pruning until subjective satisfaction.

We utilize the same approximately 1M parameter NVP model from earlier experiments, with an initial count of 20 pseudo-inputs initialized randomly. The model is trained for 100 epochs with a maximum learning rate of $0.003$ which decays as training goes on. Training uses batches of size $256$, with a warmup period of upwards-ramping learning rate.

Initial observations are comprised of considering both the generated psuedo-inputs and samples from the actual data distribution, as can be seen in figures 3.4 and 3.3 respectively. The data samples give us the semantic bias that we are looking at clothes in a certain particular format, priming us for interpreting the pseduo-inputs. When looking at the pseudo-inputs, we can immediately tell that the model has picked up on a certain level of fidelity it finds appropriate (generally this is bounded by the capacity of the model, as well as the distinction of the underlying data). In this case, we note that the general categories that the pseudo-inputs have aligned with are: pants (7, 9, 12), short-sleeved shirts (19), long-sleeved shirts (2, 8, 11, 16), flat shoes (3, 4, 5), heeled shoes (17, 18, 20), purses (1, 13, 14), and dresses (6, 10). Viewing it more coarsely, one may choose not to discern between shirts regardless of sleeve length, or perhaps shoes regardless of structure, in which case we arrive at the FMNIST-5 re-interpretation of the Fashion-MNIST dataset [57].

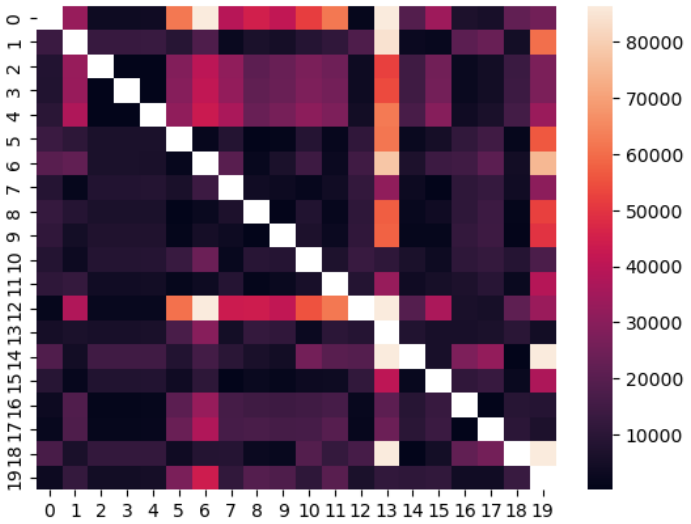


Figure 4.6: A heatmap of the KL-divergences between ordered pairs of pseudo-input posteriors. The rows indicate the first argument to KL-divergence.

From here, we consider the entropy of the batch-wise entropy of the model. At the end of training, we have $\mathbb{H}_b \approx 2.23$ meaning that we can, as a rough heuristic, expect at least $\lfloor \exp(\mathbb{H}_b) \rfloor = 9$ pseduo-inputs that the model views as necessary. Note that were the model trained longer, or with higher capacity, it is likely it would rely on *fewer* pseudo-inputs since batch-wise entropy starts maximal (in the case of 20 pseudo-inputs, we have that initially $\mathbb{H}_b = \ln(20) \approx 2.99$.

Given this heuristic, we now consider the statistical distances between points. For this case study, we use the symmetric JS-divergence. We can see the heatmap of KL-divergences in figure 4.6, which is then symmetrized to a heatmap of JS-divergences in figure 4.7. Note that the diagonals are empty since they are trivially zero and are not relevant to this procedure. From this heatmap, we can find the 11 pseudo-inputs that ought to be trimmed according to our entropy estimate.As mentioned earlier, given a pair of pseudo-inputs $(x_i^\eta, x_j^\eta)$, we remove the one that produces the lower KL-divergence when in the first argument, corresponding to the more "redundant" input with respect to its partner. Using this process, we prune the pseudo-inputs in figure 3.4 down to the nine pictured in figure 4.8.

Note that these correspond well to the five distinct classes of FMNIST-5, and are roughly proportional to, and representative of, the full Fashion-MNIST label scheme. These nine pseudo-inputs represent a *learned* surrogate sample of

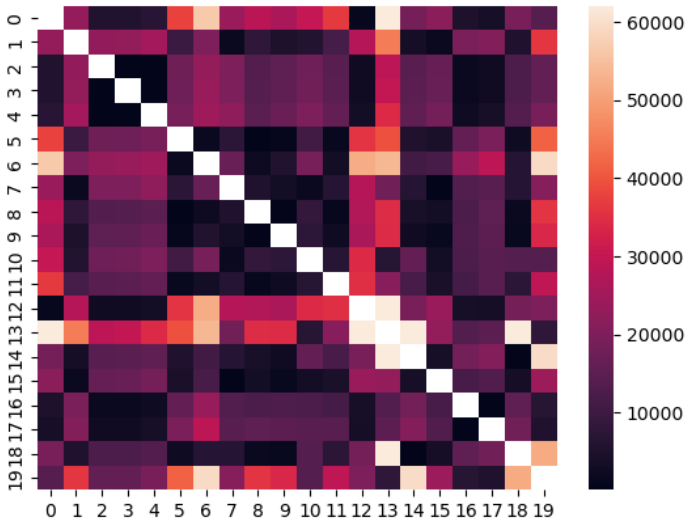


Figure 4.7: A heatmap of the JS-divergences between ordered pairs of pseudo-input posteriors.

the entire dataset, and can be generated with no expert intervention. Should an expert have a preference, they may have chosen to prune fewer, or more pseudos than the heuristic recommends. Ultimately, this depends on the level of fidelity sought after by the expert.

Thus, we demonstrate that the data-like pseudo-inputs under NVP are viable surrogate samples to inspect as a means of gaining generalizable understanding over the dataset as a whole, and we demonstrate a heuristic-driven process that can be customized by expert intervention to prune the pseudo-inputs into a representative sample. Such a method can greatly improve the development of a label schema by allowing experts to get an at-a-glance view at the diversity and distribution of points across a complex dataset with trivial examples that are simpler to analyze. The most important part is that the majority of this process is entirely unsupervised, requiring no input whatsoever.

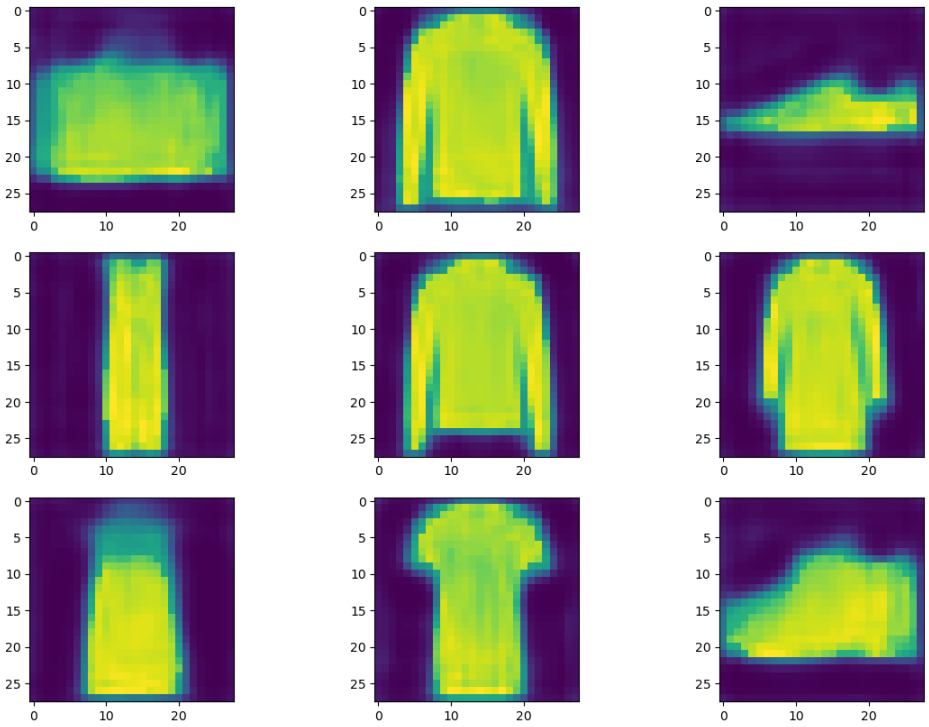

Figure 4.8: The nine remaining pseudo-inputs after the heuristic pruning process.

# Chapter 5

# Conclusion

In this dissertation, we have formalized the concept of "label schema discovery" as a problem context and proposed a series of novel contributions built on an established scientific basis to extend a new framework which allows for an accelerated, minimal-assumption workflow, combining automated geometric insights from the data with poignant and curated expert feedback. The methodology is designed around and validated on lightweight machine learning models or lightweight deep learning models, demonstrating that the efficacy of framework does not lie in over-complication and over-parameterization, but rather in the fundamental viability of the underlying approach.

We introduced PCH, a novel algorithm built atop the SOTA HDBSCAN algorithm, allowing for fast, well-scaling, semi-supervised hierarchical clustering. We demonstrate its efficacy over a standard suite of benchmark datasets used for validating conventional clustering algorithms, and even demonstrate its superiority to the practice-standard implementation of COP-Kmeans and the HDBSCAN original semi-supervised algorithm as suggested by Campello et al. [10]. We demonstrate PCH's flexibility when it comes to honing in on arbitrary levels of label fidelity by evaluating over the Anuran Calls dataset, which is organized with a hierarchical label schema, allowing for explicit comparison.

Next, we developed NVP, a novel algorithm which is an extension to a popular and commonly used VAE implementation. We demonstrate NVP's unmatched capacity to generate synthetic data to act as representative samples of the underlying dataset. We then disentangled the learned geometric manifold, and learned distribution, to allow for improved model efficacy and representation quality. We further developed a pushforward metric parameterized by the two-stage VAE implementation to allow for explicit likelihood calculations of the data under the ancestral sampling process. We validated the efficacy of NVP against a standard VAE and demonstrate its learning capacity increase. We then

inspected the pseudo-inputs of the NVP and compare against real data, showing that even at tiny model capacities, we are able to arrive at interpretable and usable pseudo-inputs. We also introduced LSV, which builds atop the NVP framework to further localize spatial regions in the learned geometric manifold. We demonstrated its learning capacity by directly considering the labels obtained by MLE estimation over a subset of its pseudo-inputs and compare it against both unsupervised, and semi-supervised methods.

Finally, we proved that the pushforward metric induced by the LSV can be used as a stand-in for a distribution likelihood function over a discrete dataset without the need for any explicit parameterizations and distribution assumptions, in effect unlocking a suite of active learning methods for otherwise-deterministic methods such as PCH. We then introduce the radial sampling method for iterative improvement of the underlying PCH model, developing an active-learning scheme specialized for the algorithm itself. We demonstrated the efficacy of radial sampling against uniform sampling, and note its informative nature. Finally, we proposed a workflow for pseudo-input introspection so as make the most out of the LSV model. We demonstrated through a case study that the workflow is easy, mostly automated, and able to include expert sentiment explicitly through determining how many pseudo-inputs to prune, effectively selecting the level of label fidelity in a globally consistent way. We furthermore demonstrated that an expert need not intervene, and that information-theoretic heuristics can be used to fully automate the process while still delivering high quality results.

Thus, we provided a complete system of tools for data representation, semi-supervised clustering, high-efficacy querying, and robust data introspection for label schema discovery.

# Bibliography


[1] Kofi P. Adragni and R. Dennis Cook. "Sufficient dimension reduction and prediction in regression". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367.1906 (Nov. 2009). Publisher: Royal Society, pp. 4385–4405. DOI: 10.1098/rsta.2009.0110. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2009.0110 (visited on 03/11/2025).

[2] A. Asperti, D. Evangelista, and E. Loli Piccolomini. *A survey on Variational Autoencoders from a GreenAI perspective*. arXiv:2103.01071 [cs]. Mar. 2021. DOI: 10.48550/arXiv.2103.01071. URL: http://arxiv.org/abs/2103.01071 (visited on 03/11/2025).

[3] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. "Overview and comparative study of dimensionality reduction techniques for high dimensional data". In: *Information Fusion* 59 (July 2020), pp. 44–58. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2020.01.005. URL: https://www.sciencedirect.com/science/article/pii/S156625351930377X (visited on 03/11/2025).

[4] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. "Active Semi-Supervision for Pairwise Constrained Clustering". en. In: *Proceedings of the 2004 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2004, pp. 333–344. ISBN: 978-0-89871-568-2 978-1-61197-274-0. DOI: 10.1137/1.9781611972740.31. URL: https://epubs.siam.org/doi/10.1137/1.9781611972740.31 (visited on 03/10/2025).

[5] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering". In: *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press, 2001. URL: https://proceedings.neurips.cc/paper_files/paper/2001/hash/f106b7f99d2cb30c3db1c3cc0fde9ccb-Abstract.html (visited on 09/30/2024).

[6] Yoav Benjamini and Daniel Yekutieli. "The control of the false discovery rate in multiple testing under dependency". In: *The Annals of Statistics* 29.4 (Aug. 2001). Publisher: Institute of Mathematical Statistics, pp. 1165–1188. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1013699998. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-4/The-control-of-the-false-discovery-rate-in-multiple-testing/10.1214/aos/1013699998.full (visited on 03/23/2025).

[7] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. "Integrating constraints and metric learning in semi-supervised clustering". en. In: *Twenty-first international conference on Machine learning - ICML '04*. Banff, Alberta, Canada: ACM Press, 2004, p. 11. DOI: 10.1145/1015330.1015360. URL: http://portal.acm.org/citation.cfm?doid=1015330.1015360 (visited on 03/10/2025).

[8] Jianghui Cai et al. "A review on semi-supervised clustering". en. In: *Information Sciences* 632 (June 2023), pp. 164–200. ISSN: 00200255. DOI: 10.1016/j.ins.2023.02.088. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020025523002840 (visited on 01/14/2025).

[9] Tadeusz Caliński and Harabasz JA. "A Dendrite Method for Cluster Analysis". In: *Communications in Statistics - Theory and Methods* 3 (Jan. 1974), pp. 1–27. DOI: 10.1080/03610927408827101.

[10] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". en. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Berlin, Heidelberg: Springer, 2013, pp. 160–172. ISBN: 978-3-642-37456-2. DOI: 10.1007/978-3-642-37456-2_14.

[11] Ricardo J. G. B. Campello et al. "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection". In: *ACM Trans. Knowl. Discov. Data* 10.1 (July 2015), 5:1–5:51. ISSN: 1556-4681. DOI: 10.1145/2733381. URL: https://doi.org/10.1145/2733381 (visited on 03/07/2025).

[12] Juan Sebastián Cañas et al. "A dataset for benchmarking Neotropical anuran calls identification in passive acoustic monitoring". en. In: *Scientific Data* 10.1 (Nov. 2023). Publisher: Nature Publishing Group, p. 771. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02666-2. URL: https://www.nature.com/articles/s41597-023-02666-2 (visited on 03/18/2025).

[13] Mathilde Caron et al. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. arXiv:2006.09882 [cs]. Jan. 2021. DOI: 10.48550/arXiv.2006.09882. URL: http://arxiv.org/abs/2006.09882 (visited on 03/14/2025).

[14] Yanwen Chong et al. "Graph-based semi-supervised learning: A review". en. In: *Neurocomputing* 408 (Sept. 2020), pp. 216–230. ISSN: 09252312. DOI: 10.1016/j.neucom.2019.12.130. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231220304938 (visited on 03/10/2025).

[15] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps". en. In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30. ISSN: 10635203. DOI: 10.1016/j.acha.2006.04.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S1063520306000546 (visited on 03/11/2025).

[16] Trevor Cox and Michael Cox. *Multidimensional Scaling*. 2nd ed. New York: Chapman and Hall/CRC, Sept. 2000. ISBN: 978-0-367-80170-0. DOI: 10.1201/9780367801700.

[17] John P Cunningham and Zoubin Ghahramani. "Linear Dimensionality Reduction: Survey, Insights, and Generalizations". en. In: ().

[18] Bin Dai and David Wipf. *Diagnosing and Enhancing VAE Models*. arXiv:1903.05789 [cs, stat]. Oct. 2019. DOI: 10.48550/arXiv.1903.05789. URL: http://arxiv.org/abs/1903.05789 (visited on 09/11/2024).

[19] David L. Davies and Donald W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (Apr. 1979). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 224–227. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1979.4766909. URL: https://ieeexplore.ieee.org/document/4766909 (visited on 03/28/2025).

[20] Carl Eckart and Gale Young. "The Approximation of One Matrix by Another of Lower Rank". en. In: *Psychometrika* 1.3 (Sept. 1936), pp. 211–218. ISSN: 0033-3123, 1860-0980. DOI: 10.1007/BF02288367. URL: https://www.cambridge.org/core/product/identifier/S0033312300051085/type/journal_article (visited on 03/11/2025).

[21] Michael B. Eisen et al. "Cluster analysis and display of genome-wide expression patterns". In: *Proceedings of the National Academy of Sciences* 95.25 (Dec. 1998). Publisher: Proceedings of the National Academy of Sciences, pp. 14863–14868. DOI: 10.1073/pnas.95.25.14863. URL: https://www.pnas.org/doi/10.1073/pnas.95.25.14863 (visited on 03/11/2025).

[22] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. "Testing the manifold hypothesis". en. In: *Journal of the American Mathematical Society* 29.4 (Oct. 2016), pp. 983–1049. ISSN: 0894-0347, 1088-6834. DOI: 10.1090/jams/852. URL: https://www.ams.org/jams/2016-29-04/S0894-0347-2016-00852-4/ (visited on 09/13/2024).

[23] R. A. Fisher. "The Use of Multiple Measurements in Taxonomic Problems". en. In: *Annals of Eugenics* 7.2 (1936). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x, pp. 179–188. ISSN: 2050-1439. DOI: 10.1111/j.1469-1809.1936.tb02137.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x (visited on 03/11/2025).

[24] I K Fodor. *A Survey of Dimension Reduction Techniques*. en. Tech. rep. UCRL-ID-148494, 15002155. May 2002, UCRL–ID–148494, 15002155. DOI: 10.2172/15002155. URL: http://www.osti.gov/servlets/purl/15002155-mumfPN/native/ (visited on 03/11/2025).

[25] Atsushi Fujii et al. "Selective Sampling for Example-based Word Sense Disambiguation". In: *Computational Linguistics* 24.4 (1998). Ed. by Julia Hirschberg. Place: Cambridge, MA Publisher: MIT Press, pp. 573–597. URL: https://aclanthology.org/J98-4002/ (visited on 03/15/2025).

[26] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. *Dimensionality Reduction for Supervised Learning With Reproducing Kernel Hilbert Spaces:* en. Tech. rep. Fort Belvoir, VA: Defense Technical Information Center, May 2003. DOI: 10.21236/ADA446572. URL: https://apps.dtic.mil/sti/citations/tr/ADA446572 (visited on 03/11/2025).

[27] Jie Gui et al. *A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends*. arXiv:2301.05712 [cs]. July 2024. DOI: 10.48550/arXiv.2301.05712. URL: http://arxiv.org/abs/2301.05712 (visited on 03/14/2025).

[28] Alexander G. Hauptmann et al. "Extreme video retrieval: joint maximization of human and computer performance". In: *Proceedings of the 14th ACM international conference on Multimedia*. MM '06. New York, NY, USA: Association for Computing Machinery, Oct. 2006, pp. 385–394. ISBN: 978-1-59593-447-5. DOI: 10.1145/1180639.1180721. URL: https://doi.org/10.1145/1180639.1180721 (visited on 03/15/2025).

[29] Kaiming He et al. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385 (visited on 03/19/2025).

[30] E. Hellinger. "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen." de. In: *Journal für die reine und angewandte Mathematik* 1909.136 (July 1909). Publisher: De Gruyter, pp. 210–271. ISSN: 1435-5345. DOI: 10.1515/crll.1909.136.210. URL: https://www.degruyter.com/document/doi/10.1515/crll.1909.136.210/html (visited on 03/17/2025).

[31] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html (visited on 03/17/2025).

[32] Matthew D Hoffman and Matthew J Johnson. "ELBO surgery: yet another way to carve up the variational evidence lower bound". en. In: ().

[33] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. "Large-scale text categorization by batch mode active learning". In: *Proceedings of the 15th international conference on World Wide Web*. WWW '06. New York, NY, USA: Association for Computing Machinery, May 2006, pp. 633–642. ISBN: 978-1-59593-323-2. DOI: 10.1145/1135777.1135870. URL: https://doi.org/10.1145/1135777.1135870 (visited on 03/15/2025).

[34] Harold Hotelling. "Relations Between Two Sets of Variates". In: *Biometrika* 28.3/4 (1936). Publisher: [Oxford University Press, Biometrika Trust], pp. 321–377. ISSN: 0006-3444. DOI: 10.2307/2333955. URL: https://www.jstor.org/stable/2333955 (visited on 03/11/2025).

[35] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. en. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075. URL: https://doi.org/10.1007/BF01908075 (visited on 03/08/2025).

[36] Dino Ienco et al. “Clustering Based Active Learning for Evolving Data Streams”. en. In: *Discovery Science*. Ed. by Johannes Fürnkranz, Eyke Hüllermeier, and Tomoyuki Higuchi. Berlin, Heidelberg: Springer, 2013, pp. 79–93. ISBN: 978-3-642-40897-7. DOI: 10.1007/978-3-642-40897-7_6.

[37] Ashish Jaiswal et al. “A Survey on Contrastive Self-Supervised Learning”. en. In: *Technologies* 9.1 (Mar. 2021). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, p. 2. ISSN: 2227-7080. DOI: 10.3390/technologies9010002. URL: https://www.mdpi.com/2227-7080/9/1/2 (visited on 03/14/2025).

[38] Weikuan Jia et al. “Feature dimensionality reduction: a review”. en. In: *Complex & Intelligent Systems* 8.3 (June 2022), pp. 2663–2693. ISSN: 2198-6053. DOI: 10.1007/s40747-021-00637-x. URL: https://doi.org/10.1007/s40747-021-00637-x (visited on 03/11/2025).

[39] Jaeho Kang, Kwang Ryel Ryu, and Hyuk-Chul Kwon. “Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification”. en. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang. Berlin, Heidelberg: Springer, 2004, pp. 384–388. ISBN: 978-3-540-24775-3. DOI: 10.1007/978-3-540-24775-3_46.

[40] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. arXiv:2001.08361 [cs]. Jan. 2020. DOI: 10.48550/arXiv.2001.08361. URL: http://arxiv.org/abs/2001.08361 (visited on 03/13/2025).

[41] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. arXiv:1312.6114 [stat]. Dec. 2022. DOI: 10.48550/arXiv.1312.6114. URL: http://arxiv.org/abs/1312.6114 (visited on 03/11/2025).

[42] Diederik P. Kingma et al. *Improving Variational Inference with Inverse Autoregressive Flow*. arXiv:1606.04934 [cs, stat]. Jan. 2017. DOI: 10.48550/arXiv.1606.04934. URL: http://arxiv.org/abs/1606.04934 (visited on 09/27/2024).

[43] Brian Kulis. “Metric Learning: A Survey”. en. In: *Foundations and Trends® in Machine Learning* 5.4 (2013), pp. 287–364. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000019. URL: http://www.nowpublishers.com/articles/foundations-and-trends-in-machine-learning/MAL-019 (visited on 03/11/2025).

[44] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (Mar. 1951). Publisher: Institute of Mathematical Statistics, pp. 79–86. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729694. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full (visited on 03/12/2025).

[45] Daniel D. Lee and H. Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. en. In: *Nature* 401.6755 (Oct. 1999). Publisher: Nature Publishing Group, pp. 788–791. ISSN: 1476-4687. DOI: 10.1038/44565. URL: https://www.nature.com/articles/44565 (visited on 03/11/2025).

[46] David D. Lewis and Jason Catlett. “Heterogeneous Uncertainty Sampling for Supervised Learning”. en. In: *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 148–156. ISBN: 978-1-55860-335-6. DOI: 10.1016/B978-1-55860-335-6.50026-X. URL: https://linkinghub.elsevier.com/retrieve/pii/B978155860335650026X (visited on 03/15/2025).

[47] David D. Lewis and William A. Gale. “A Sequential Algorithm for Training Text Classifiers”. en. In: *SIGIR ’94*. Ed. by Bruce W. Croft and C. J. van Rijsbergen. London: Springer, 1994, pp. 3–12. ISBN: 978-1-4471-2099-5. DOI: 10.1007/978-1-4471-2099-5_1.

[48] Dongyuan Li et al. *A Survey on Deep Active Learning: Recent Advances and New Frontiers*. arXiv:2405.00334 [cs]. July 2024. DOI: 10.48550/arXiv.2405.00334. URL: http://arxiv.org/abs/2405.00334 (visited on 03/14/2025).

[49] Jianhua Lin. “Divergence Measures Based on the Shannon Entropy”. en. In: ().

[50] Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. “Selective Sampling for Nearest Neighbor Classifiers”. en. In: *Machine Learning* 54.2 (Feb. 2004), pp. 125–152. ISSN: 1573-0565. DOI: 10.1023/B:MACH.0000011805.60520.fe. URL: https://doi.

org/10.1023/B:MACH.0000011805.60520.fe (visited on 03/15/2025).

[51] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. ISSN: 1533-7928. URL: http://jmlr.org/papers/v9/vandermaaten08a.html (visited on 09/27/2024).

[52] Alireza Makhzani et al. *Adversarial Autoencoders*. arXiv:1511.05644 [cs]. May 2016. DOI: 10.48550/arXiv.1511.05644. URL: http://arxiv.org/abs/1511.05644 (visited on 03/12/2025).

[53] Andrew Kachites McCallum and Kamal Nigam. "Employing EM and Pool-Based Active Learning for Text Classification". en. In: ().

[54] Leland McInnes and John Healy. "Accelerated Hierarchical Density Clustering". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. arXiv:1705.07321 [stat]. Nov. 2017, pp. 33–42. DOI: 10.1109/ICDMW.2017.12. URL: http://arxiv.org/abs/1705.07321 (visited on 03/07/2025).

[55] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv:1802.03426 [cs, stat]. Sept. 2020. DOI: 10.48550/arXiv.1802.03426. URL: http://arxiv.org/abs/1802.03426 (visited on 09/30/2024).

[56] Shervin Minaee et al. *Large Language Models: A Survey*. arXiv:2402.06196 [cs]. Feb. 2024. DOI: 10.48550/arXiv.2402.06196. URL: http://arxiv.org/abs/2402.06196 (visited on 03/11/2025).

[57] Sudipto Mukherjee et al. *ClusterGAN : Latent Space Clustering in Generative Adversarial Networks*. arXiv:1809.03627 [cs]. Jan. 2019. DOI: 10.48550/arXiv.1809.03627. URL: http://arxiv.org/abs/1809.03627 (visited on 03/18/2025).

[58] Abu Quwsar Ohi et al. "AutoEmbedder: A semi-supervised DNN embedding system for clustering". In: *Knowledge-Based Systems* 204 (Sept. 2020), p. 106190. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2020.106190. URL: https://www.sciencedirect.com/science/article/pii/S0950705120304172 (visited on 03/11/2025).

[59] Utku Ozbulak et al. "Know Your Self-supervised Learning: A Survey on Image-based Generative and Discriminative Training". en. In: *Transactions on Machine Learning Research* (Feb. 2023). ISSN: 2835-8856. URL: https://openreview.net/forum?id=Ma25S4ludQ (visited on 03/14/2025).

[60] A. Cerdeira Paulo Cortez. *Wine Quality*. 2009. DOI: 10.24432/C56S3T. URL: https://archive.ics.uci.edu/dataset/186 (visited on 03/18/2025).

[61] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14786440109462720, pp. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720. URL: https://doi.org/10.1080/14786440109462720 (visited on 03/11/2025).

[62] Pengjiang Qian et al. "Affinity and Penalty Jointly Constrained Spectral Clustering With All-Compatibility, Flexibility, and Robustness". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.5 (May 2017). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 1123–1138. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2015.2511179. URL: https://ieeexplore.ieee.org/document/7412775/?arnumber=7412775 (visited on 03/10/2025).

[63] C. Radhakrishna Rao. "The Utilization of Multiple Measurements in Problems of Biological Classification". en. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 10.2 (July 1948), pp. 159–193. ISSN: 1369-7412, 1467-9868. DOI: 10.1111/j.2517-6161.1948.tb00008.x. URL: https://academic.oup.com/jrsssb/article/10/2/159/7026550 (visited on 03/11/2025).

[64] Danilo Jimenez Rezende and Shakir Mohamed. *Variational Inference with Normalizing Flows*. arXiv:1505.05770 [stat]. June 2016. DOI: 10.48550/arXiv.1505.05770. URL: http://arxiv.org/abs/1505.05770 (visited on 03/12/2025).

[65] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv:2112.10752 [cs]. Apr. 2022. DOI: 10.48550/arXiv.2112.10752. URL: http://arxiv.org/abs/2112.10752 (visited on 03/11/2025).

[66] Andrew Rosenberg and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Ed. by Jason Eisner. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 410–420. URL: https://aclanthology.org/D07-1043/ (visited on 03/28/2025).

[67] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. ISSN: 0377-0427. DOI: 10.1016/0377-0427(87)90125-7. URL: https://www.sciencedirect.com/science/article/pii/0377042787901257 (visited on 03/28/2025).

[68] Sam T. Roweis and Lawrence K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear Embedding". en. In: *Science* 290.5500 (Dec. 2000), pp. 2323–2326. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.290.5500.2323. URL: https://www.science.org/doi/10.1126/science.290.5500.2323 (visited on 03/11/2025).

[69] Friedhelm Schwenker and Edmondo Trentin. "Pattern classification and clustering: A review of partially supervised learning approaches". en. In: *Pattern Recognition Letters* 37 (Feb. 2014), pp. 4–14. ISSN: 01678655. DOI: 10.1016/j.patrec.2013.10.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167865513004091 (visited on 03/10/2025).

[70] Eran Segal et al. "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data". eng. In: *Nature Genetics* 34.2 (June 2003), pp. 166–176. ISSN: 1061-4036. DOI: 10.1038/ng1165.

[71] Ozan Sener and Silvio Savarese. *Active Learning for Convolutional Neural Networks: A Core-Set Approach*. arXiv:1708.00489 [stat]. June 2018. DOI: 10.48550/arXiv.1708.00489. URL: http://arxiv.org/abs/1708.00489 (visited on 03/15/2025).

[72] Burr Settles. "Active Learning Literature Survey". en. In: ().

[73] C. E. Shannon. "A Mathematical Theory of Communication". en. In: *Bell System Technical Journal* 27.3 (1948). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x, pp. 379–423. ISSN: 1538-7305. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x (visited on 03/28/2025).

[74] Benjamin W.B. Shires and Chris J. Pickard. "(PDF) Visualizing Energy Landscapes through Manifold Learning". en. In: *ResearchGate* (Dec. 2024). DOI: 10.1103/PhysRevX.11.041026. URL: https://www.researchgate.net/publication/355950962_Visualizing_Energy_Landscapes_through_Manifold_Learning (visited on 03/06/2025).

[75] Aman Singh and Tokunbo Ogunfunmi. "An Overview of Variational Autoencoders for Source Separation, Finance, and Bio-Signal Applications". In: *Entropy* 24.1 (Dec. 2021), p. 55. ISSN: 1099-4300. DOI: 10.3390/e24010055. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8774760/ (visited on 09/27/2024).

[76] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. *A survey of dimensionality reduction techniques*. arXiv:1403.2877 [stat]. Mar. 2014. DOI: 10.48550/arXiv.1403.2877. URL: http://arxiv.org/abs/1403.2877 (visited on 03/11/2025).

[77] C. Spearman. "'General intelligence,' objectively determined and measured". In: *The American Journal of Psychology* 15.2 (1904). Place: US Publisher: Univ of Illinois Press, pp. 201–293. ISSN: 1939-8298. DOI: 10.2307/1412107.

[78] J. B. Tenenbaum, V. de Silva, and J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction". eng. In: *Science (New York, N.Y.)* 290.5500 (Dec. 2000), pp. 2319–2323. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2319.

[79] Alaa Tharwat and Wolfram Schenck. "A Survey on Active Learning: State-of-the-Art, Practical Challenges and Research Directions". en. In: *Mathematics* 11.4 (Jan. 2023). Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 820. ISSN: 2227-7390. DOI: 10.3390/math11040820. URL: https://www.mdpi.com/2227-7390/11/4/820 (visited on 03/14/2025).

[80] Eric K. Tokuda, Cesar H. Comin, and Luciano Da F. Costa. "Revisiting agglomerative clustering". en. In: *Physica A: Statistical Mechanics and its Applications* 585 (Jan. 2022), p. 126433. ISSN: 03784371. DOI: 10.1016/j.physa.2021.126433. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378437121007068 (visited on 03/10/2025).

[81] Jakub M. Tomczak and Max Welling. *Improving Variational Auto-Encoders using Householder Flow*. arXiv:1611.09630 [cs]. Jan. 2017. DOI: 10.48550/arXiv.1611.09630. URL: http://arxiv.org/abs/1611.09630 (visited on 03/12/2025).

[82] Jakub M. Tomczak and Max Welling. *VAE with a VampPrior*. arXiv:1705.07120 [cs, stat]. Feb. 2018. DOI: 10.48550/arXiv.1705.07120. URL: http://arxiv.org/abs/1705.07120 (visited on 09/11/2024).

[83] Simon Tong and Edward Chang. "Support vector machine active learning for image retrieval". In: *Proceedings of the ninth ACM international conference on Multimedia*. MULTIMEDIA '01. New York, NY, USA: Association for Computing Machinery, Oct. 2001, pp. 107–118. ISBN: 978-1-58113-394-3. DOI: 10.1145/500141.500159. URL: https://doi.org/10.1145/500141.500159 (visited on 03/15/2025).

[84] Simon Tong and Daphne Koller. "Support Vector Machine Active Learning with Applications to Text Classification". en. In: ().

[85] Warren S. Torgerson. "Multidimensional scaling: I. Theory and method". en. In: *Psychometrika* 17.4 (Dec. 1952), pp. 401–419. ISSN: 1860-0980. DOI: 10.1007/BF02288916. URL: https://doi.org/10.1007/BF02288916 (visited on 03/11/2025).

[86] Duy Tin Truong and Roberto Battiti. *A Survey of Semi-Supervised Clustering Algorithms: from a priori scheme to interactive scheme and open issues*. Departmental Technical Report. Trento: University of Trento, July 2013. URL: http://eprints.biblio.unitn.it/4198/ (visited on 03/10/2025).

[87] Gokhan Tur, Dilek Hakkani-Tür, and Robert E. Schapire. "Combining active and semi-supervised learning for spoken language understanding". In: *Speech Communication* 45.2 (Feb. 2005), pp. 171–186. ISSN: 0167-6393. DOI: 10.1016/j.specom.2004.08.002. URL: https://www.sciencedirect.com/science/article/pii/S0167639304000962 (visited on 03/15/2025).

[88] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *Advances in Neural Information Processing Systems*. Vol. 18. MIT Press, 2005. URL: https://papers.nips.cc/paper_files/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html (visited on 03/11/2025).

[89] Yi Wu et al. "Sampling Strategies for Active Learning in Personal Photo Retrieval". In: *2006 IEEE International Conference on Multimedia and Expo*. ISSN: 1945-788X. July 2006, pp. 529–532. DOI: 10.1109/ICME.2006.262442. URL: https://ieeexplore.ieee.org/document/4036653 (visited on 03/15/2025).

[90] I. Xenarios et al. "DIP: The Database of Interacting Proteins: 2001 update". eng. In: *Nucleic Acids Research* 29.1 (Jan. 2001), pp. 239–241. ISSN: 1362-4962. DOI: 10.1093/nar/29.1.239.

[91] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:1708.07747 [cs]. Sept. 2017. DOI: 10.48550/arXiv.1708.07747. URL: http://arxiv.org/abs/1708.07747 (visited on 03/18/2025).

[92] Yan, Jie Yang, and Hauptmann. "Automatically labeling video data using multi-class active learning". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. Oct. 2003, 516–523 vol.1. DOI: 10.1109/ICCV.2003.1238391. URL: https://ieeexplore.ieee.org/document/1238391 (visited on 03/15/2025).

[93] Haifeng Yang et al. "Data mining techniques on astronomical spectra data – I. Clustering analysis". In: *Monthly Notices of the Royal Astronomical Society* 517.4 (Dec. 2022), pp. 5496–5523. ISSN: 0035-8711. DOI: 10.1093/mnras/stac2975. URL: https://doi.org/10.1093/mnras/stac2975 (visited on 03/10/2025).

[94] Liu Yang and Rong Jin. "Distance Metric Learning: A Comprehensive Survey". en. In: ().

[95] Xuesong Yin, Ting Shu, and Qi Huang. "Semi-supervised fuzzy clustering with metric learning and entropy regularization". en. In: *Knowledge-Based Systems* 35 (Nov. 2012), pp. 304–311. ISSN: 09507051. DOI: 10.1016/j.knosys.2012.05.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705112001669 (visited on 03/10/2025).

[96] Xuesong Yin et al. "Semi-supervised clustering with metric learning: An adaptive kernel method". In: *Pattern Recognition* 43.4 (Apr. 2010), pp. 1320–1333. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2009.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0031320309004191 (visited on 03/10/2025).

[97] Stella X Yu and Jianbo Shi. "Segmentation Given Partial Grouping Constraints". en. In: ().

[98] Meekail Zain et al. "Towards an Unsupervised Spatiotemporal Representation of Cilia Video Using A Modular Generative Pipeline". en. In: Austin, Texas, 2020, pp. 166–175. DOI: 10.25080/Majora-342d178e-017. URL: https://doi.curvenote.com/10.25080/Majora-342d178e-017 (visited on 03/13/2025).

[99] Xueying Zhan et al. *A Comparative Survey of Deep Active Learning*. arXiv:2203.13450 [cs]. July 2022. DOI: 10.48550/arXiv.2203.13450. URL: http://arxiv.org/abs/2203.13450 (visited on 09/12/2024).

[100] Cha Zhang and Tsuhan Chen. "An active learning framework for content-based information retrieval". In: *IEEE Transactions on Multimedia* 4.2 (June 2002). Conference Name: IEEE Transactions on Multimedia, pp. 260–268. ISSN: 1941-0077. DOI: 10.1109/TMM.2002.1017738. URL: https://ieeexplore.ieee.org/document/1017738 (visited on 03/15/2025).

[101] Huaxiang Zhang and Jing Lu. "Semi-supervised fuzzy clustering: A kernel-based approach". en. In: *Knowledge-Based Systems* 22.6 (Aug. 2009), pp. 477–481. ISSN: 09507051. DOI: 10.1016/j.knosys.2009.06.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705109000987 (visited on 03/10/2025).

[102] Shengjia Zhao, Jiaming Song, and Stefano Ermon. *InfoVAE: Information Maximizing Variational Autoencoders*. arXiv:1706.02262 [cs]. May 2018. DOI: 10.48550/arXiv.1706.02262. URL: http://arxiv.org/abs/1706.02262 (visited on 03/12/2025).

[103] Yuhang Zhao et al. "An independent central point OPTICS clustering algorithm for semi-supervised outlier detection of continuous glucose measurements". en. In: *Biomedical Signal Processing and Control* 71 (Jan. 2022), p. 103196. ISSN: 17468094. DOI: 10.1016/j.bspc.2021.103196. URL: https://linkinghub.elsevier.com/retrieve/pii/S174680942100793X (visited on 03/10/2025).