entropy

Article

Pareto-Optimal Clustering with the Primal Deterministic Information Bottleneck

Andrew K. Tan [1,2,*], Max Tegmark [1,2] and Isaac L. Chuang [1,2,3]

[1] Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; tegmark@mit.edu (M.T.); ichuang@mit.edu (I.L.C.)
[2] The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Cambridge, MA 02139, USA
[3] Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
* Correspondence: aktan@mit.edu

Abstract: At the heart of both lossy compression and clustering is a trade-off between the fidelity and size of the learned representation. Our goal is to map out and study the Pareto frontier that quantifies this trade-off. We focus on the optimization of the Deterministic Information Bottleneck (DIB) objective over the space of hard clusterings. To this end, we introduce the *primal* DIB problem, which we show results in a much richer frontier than its previously studied Lagrangian relaxation when optimized over discrete search spaces. We present an algorithm for mapping out the Pareto frontier of the primal DIB trade-off that is also applicable to other two-objective clustering problems. We study general properties of the Pareto frontier, and we give both analytic and numerical evidence for logarithmic sparsity of the frontier in general. We provide evidence that our algorithm has polynomial scaling despite the super-exponential search space, and additionally, we propose a modification to the algorithm that can be used where sampling noise is expected to be significant. Finally, we use our algorithm to map the DIB frontier of three different tasks: compressing the English alphabet, extracting informative color classes from natural images, and compressing a group theory-inspired dataset, revealing interesting features of frontier, and demonstrating how the structure of the frontier can be used for model selection with a focus on points previously hidden by the cloak of the convex hull.

Keywords: multi-objective; optimization; Pareto; frontier; information; bottleneck; clustering

check for updates

Citation: Tan, A.K.; Tegmark, M.; Chuang, I.L. Pareto-Optimal Clustering with the Primal Deterministic Information Bottleneck. *Entropy* **2022**, *24*, 771. https://doi.org/10.3390/e24060771

Academic Editor: Raúl Alcaraz

Received: 15 April 2022
Accepted: 24 May 2022
Published: 30 May 2022

1. Introduction

Many important machine learning tasks can be cast as an optimization of two objectives that are fundamentally in conflict: performance and parsimony. In an auto-encoder, this trade-off is between the fidelity of the reconstruction and narrowness of the bottleneck. In the rate-distortion setting, the quantities of interest are the distortion, as quantified by a prescribed distortion function, and the captured information. For clustering, the trade-off is between intra-cluster variation and the number of clusters. While these problems come in many flavors—with different motivations, domains, objectives, and solutions—what is common to all such multi-objective trade-offs is the existence of a Pareto frontier, representing the boundary separating feasible solutions from infeasible ones. In a two-objective optimization problem, this boundary is generically a one-dimensional curve in the objective plane, representing solutions to the trade-off where increasing performance along one axis necessarily decreases performance along the other.

The shape of the frontier, at least locally, is important for model selection: prominent corners on the frontier are often more robust to changes in the inputs and therefore correspond to more desirable solutions. The global frontier can provide additional insights, such as giving a sense of interesting scales in the objective function. Structure often exists at multiple scales; for hierarchical clustering problems, these are the scales at which the data naturally resolve. Unfortunately, much of this useful structure (see Figure 1) is inaccessible to optimizers of the more commonly studied convex relaxations of the trade-offs. Optimization over discrete search spaces poses a particular difficulty to convex relaxed formulations, as most points on the convex hull are not feasible solutions, and Pareto optimal solutions are masked by the hull. While the optimization of the Lagrangian relaxation is often sufficient for finding a point on or near the frontier, we, in contrast, seek to map out the entire frontier of the trade-off and therefore choose to tackle the primal problem directly.

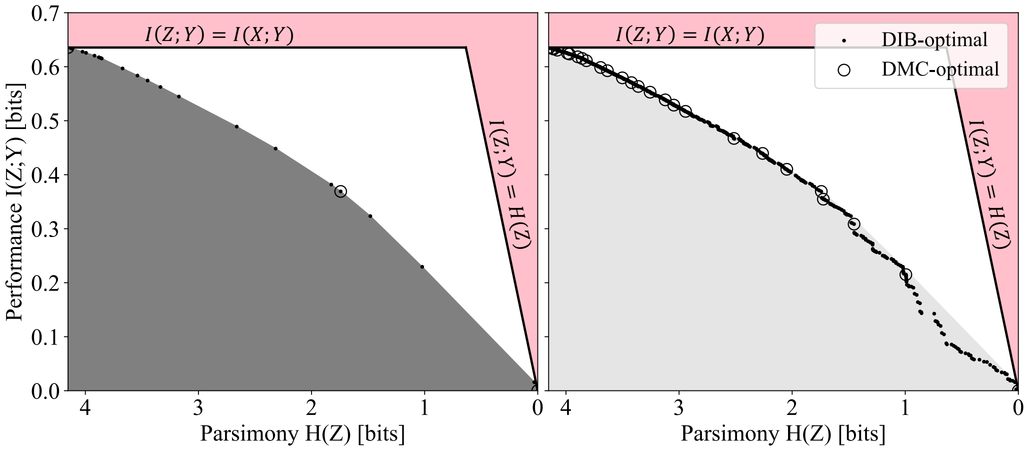


Figure 1. Comparison of the Lagrangian DIB (**left**) and primal DIB (**right**) frontiers discovered by Algorithm 1 for the English alphabet compression task discussed in Section 3.2.1. The shaded regions indicate the convex hull of the points found by the Pareto Mapper algorithm. Clusterings inside the shaded region, while Pareto optimal, are not optimal in the Lagrangian formulation.

Algorithm 1 Pareto Mapper: ε-greedy agglomerative search

```
Input: Joint distribution $X, Y \sim p_{XY}$, and search parameter $\varepsilon$
Output: Approximate Pareto frontier $P$
procedure PARETO_MAPPER($p_{XY}, \varepsilon$)
    Pareto Set $P = \emptyset$                                ▷ Initialize maintained Pareto Set
    Queue $Q = \emptyset$                                     ▷ Initialize search queue
    Point $p = (\mathrm{x} = -\mathrm{H}(p_X), \mathrm{y} = \mathrm{I}(p_{X;Y}), \mathrm{f} = \mathrm{id})$     ▷ Evaluate trivial clustering
    $P \leftarrow$ INSERT$(p, P)$
    $Q \leftarrow$ ENQUEUE(id, $Q$) ▷ Start with identity clustering id : $[n] \to [n]$ where $n = |X|$
    while $Q$ is not $\emptyset$ do
        $f =$ DEQUEUE$(Q)$
        $n = |\mathrm{range}(f)|$
        for $0 < i < j < n$ do                      ▷ Loop over all pairs of output clusters of $f$
            $f' = c_{i,j} \circ f$                   ▷ Merge clusters $i, j$ output $f$
            Point $p = \mathrm{Point}(\mathrm{x} = -\mathrm{H}(p_{f'(X)}), \mathrm{y} = \mathrm{I}(p_{f'(X);Y}), \mathrm{f} = f')$
            $d =$ PARETO_DISTANCE$(p, P)$
            $P \leftarrow$ PARETO_ADD$(p, P)$   ▷ Keep track of point and clustering in Pareto Set
            with probability $e^{-d/\varepsilon}$, $Q \leftarrow$ ENQUEUE$(f', Q)$

    return $P$
```

We focus on the general problem of the deterministic encoding of a discrete domain. For a finite set of inputs, X, which we identify with the integers $[X] \equiv \{1, \ldots, |X|\}$, we seek a mapping to a set $[Z]$, where $|Z| \le |X|$. The search space is therefore the space of functions $f : [X] \to [Z]$, which we call "encodings" or equivalently, "hard clusterings", where $Z = f(X)$ is interpreted as the cluster to which X is assigned. We evaluate the encodings using the Deterministic Information Bottleneck objective, but regardless of which objectives are chosen, we will refer to all two-objective optimization problems over the space of such functions f as "clustering problems".

The goal of this paper is to motivate the study of the Pareto frontiers to primal clustering problems and to present a practical method for their discovery.

1.1. Objectives and Relation to Prior Work

We focus on the task of lossy compression, which is a trade-off between retaining the salient features of a source and parsimony. Rate-distortion theory provides the theoretical underpinnings for studying lossy data compression of a source random variable X into a compressed representation Z [1]. In this formalism, a distortion function quantifies the dissatisfaction with a given encoding, which is balanced against the complexity of the compressed representation as measured by $I(Z;X)$. In the well-known Information Bottleneck (IB) problem [2], the goal is to preserve information about a target random variable Y as measured by the mutual information $I(Z;Y)$; the IB can be viewed as a rate-distortion problem with the Kullback–Leibler divergence, $D_{KL}(p_{Y|X}||p_{Y|Z})$, serving as the measure of distortion. In recent years, a number of similar bottlenecks have also been proposed inspired by the IB problem [3–5]. We focus on one of these bottlenecks known as the Deterministic Information Bottleneck (DIB) [3].

1.1.1. The Deterministic Information Bottleneck

In the DIB problem, we are given random variables X and Y with joint probability mass function (PMF) p_{XY}, and we would like to maximize $I(Z;Y)$ subject to a constraint on $H(Z)$. As in [3], we further restrict ourselves to the compression of discrete domains, where X, Y and Z are finite, discrete random variables. We note that DIB-optimal encodings are deterministic encodings $Z = f(X)$ [3], and we can therefore focus on searching through the space of functions $f : [X] \to [Z]$, justifying the interpretation of DIB as a clustering problem. Since the optimization is being performed over a discrete domain in this case, not all points along the frontier are achievable. Nonetheless, we define the Pareto frontier piecewise as the curve that takes on the minimum vertical value between any two adjacent points on the frontier.

Formally, given p_{XY}, the DIB problem seeks an encoding $f^* : X \to Z$ such that, $Z^* = f^*(X)$ maximizes the relevant information captured for a given entropy limit H^*:

$$f^*_{\text{primal}} \equiv \underset{f:H[f(X)]\leq H^*}{\text{argmax}} \; I\Big(Y; f(X)\Big) \tag{1}$$

We will refer to the constrained version of the DIB problem in Equation (1) as the *primal* DIB problem, to differentiate it from its more commonly studied Lagrangian form [3]:

$$f^*_{\text{Lagrangian}} \equiv \underset{f}{\text{argmax}} \, I\Big(f(X); Y\Big) - \beta H\Big(f(X)\Big) \tag{2}$$

In this form, which we call the *Lagrangian* DIB, a trade-off parameter β is used instead of the entropy bound H^* to parameterize f_* and quantify the importance of memorization relative to parsimony. The Lagrangian relaxation removes the non-linear constraint by optimizing a proxy to the original function, known as the DIB Lagrangian, but comes at the cost of being unable to access points off the convex hull of the trade-off. We note that while we use the terminology 'primal DIB' to differentiate it from its Lagrangian form, we do not study its 'dual' version in this paper.

Many algorithms have been proposed for optimizing the IB, and more recently, the DIB objectives [6]. An iterative method that generalizes the Blahut-Arimoto algorithm was proposed alongside the IB [2] and DIB [3] algorithms. For the hierarchical clustering of finite, discrete random variables X and Y using the IB objective, both divisive [7] and agglomerative [8] methods have been studied. Relationships between geometric clustering and information theoretic clustering can also be used to optimize the IB objective in certain limits [9]. More recently, methods using neural network-based variational bounds have been proposed [10]. However, despite the wealth of proposed methods for optimizing the

(D)IB, past authors [2,3,6,10,11] have focused only on the Lagrangian form of Equation (2) and are therefore unable to find convex portions of the frontier.

Frontiers of the DIB Lagrangian and primal DIB trade-offs are contrasted in Figure 1, with the shaded gray region indicating the shroud that the optimization of the Lagrangian relaxation places on the frontier (the particular frontier presented is discussed in Section 3.2.1). Points within the shaded region are not accessible to the Lagrangian formulation of the problem as they do not optimize the Lagrangian. We also note that while the determinicity of solutions is a consequence of optimizing the Lagrangian DIB [3], the convex regions of the primal DIB frontier is known to contain soft clusterings [12,13]. In our work, the restriction to hard clusterings can be seen as a part of the problem statement. Finally, we adopt the convention of flipping the horizontal axis as in [13] which more closely matches the usual interpretation of a Pareto frontier where points further up and to the right are more desirable.

1.1.2. Discrete Memoryless Channels

A closely related trade-off is that between $I(Z;Y)$ and the number of clusters $|Z|$, which has been extensively studied in the literature on the compression of discrete memoryless channels (DMCs) [6,14,15]. In Figure 1 and the other frontier plots presented in Section 3.2, the DMC optimal points are plotted as open circles. The DIB and DMC trade-offs are similar enough that they are sometimes referred to interchangeably [6]: some previous proposals for solutions to the IB [8] are better described as solutions to the DMC trade-off. We would like to make this distinction explicit, as we seek to demonstrate the richness of the DIB frontier over that of the DMC frontier.

1.1.3. Pareto Front Learning

In recent work by Navon et al. [16], the authors define the problem of Pareto Front Learning (PFL) as the task of discovering the Pareto frontier in a multi-objective optimization problem, allowing for a model to be selected from the frontier at runtime. Recent hypernetwork-based approaches to PFL [16,17] are promising being both scalable and in principle capable of discovering the entirety of the primal frontier. Although we use a different approach, our work can be seen as an extension to the existing methods for PFL to discrete search spaces. Our Pareto Mapper algorithm performs PFL for the task of hard clustering, and our analysis provides evidence for the tractability of the PFL in this setting.

We also note similarities to the problems of Multi-Task Learning and Multi-Objective Optimization. The main difference between these tasks and the PFL setup is the ability to select Pareto-optimal models at runtime. We direct the reader to [16], which provides a more comprehensive overview of recent work on these related problems.

1.1.4. Motivation and Objectives

Our work is, in spirit, a generalization of [13], which demonstrated a method for mapping out the primal DIB frontier for the special case of binary classification (i.e., $|Y| = 2$). Although we deviate from their assumptions, assuming that X is discrete (rather than continuous in [13]), and being limited to deterministic encodings (rather than stochastic ones in [13]), and thus our results are not strictly comparable, our goal of mapping out the primal Pareto frontier is done in the same spirit.

The most immediate motivation for mapping out the primal Pareto frontier is that its shape is useful for model selection: given multiple candidate solutions, each being near the frontier, we would often like to be able to privilege one over the others. For example, one typically favors the points that are the most robust to input noise, that is, those that are most separated from their neighbors, appearing as concave corners in the frontier. For the problem of geometric clustering with the Lagrangian DIB, the angle between piecewise linear portions of its frontier, known as the "kink angle", has been proposed as a criterion for model selection [18]. Using the primal DIB frontier, we can use distance from the frontier as a sharper criterion for model selection; this is particularly evident in the example discussed in Section 3.2.3, where the most natural solutions are clearly prominent in the primal frontier but have zero kink angle. The structure of this frontier also encodes useful information about the data. For clustering, corners in this frontier often indicate scales of interests: those at which the data best resolve into distinct clusters. Determining these scales is the goal of hierarchical clustering.

Unlike the previously studied case of binary classification [13], no polynomial time algorithm is known for finding optimal clusterings for general $|Y| > 2$ [15]. Finding an optimal solution to the DIB problem (i.e., one point near the frontier) is known to be equivalent to k-means in certain limits [18,19], which is itself an NP-hard problem [20]: mapping out the entirety of the frontier is no easier. More fundamentally, the number of possible encoders is known to grow super-exponentially with $|X|$; therefore, it is not known whether we can even hope to store an entire DIB frontier in memory. Another issue is that of the generalization of the frontier in the presence of sampling noise. Estimation of mutual information and entropy for finite datasets is known to be a difficult problem with many common estimators either being biased, having high variance, or both [21–25]. This issue is of particular significance in our case as a noisy point on the objective plane can mask other, potentially more desirable, clusterings.

It is these gaps in the optimization of DIB and DIB-like objectives that we seek to address. Firstly, existing work on optimization concerns itself only with finding a point on or near the frontier. These algorithms may be used to map out the Pareto frontier, but they need to be run multiple times with special care taken in sampling the constraint in order to attain the desired resolution of the frontier. Furthermore, we observe empirically that almost all of the DIB Pareto frontier is in fact convex. The majority of the existing algorithms applicable to DIB-like trade-offs optimize the Lagrangian DIB [3,10] and are therefore unable to capture the complete structure of the DIB frontier. Existing agglomerative methods [8] are implicitly solving for the related but distinct DMC frontier, which has much less structure than the DIB frontier. Finally, existing methods have assumed access to the true distribution p_{XY} or otherwise used the maximum likelihood (ML) point estimators [3,13], which are known to be biased and have high variance for entropy and mutual information, which can have a significant effect on the makeup of the frontier.

1.2. Roadmap

The rest of this paper is organized as follows. In Section 2, we tackle the issue of finding the Pareto frontier in practice by proposing a simple agglomerative algorithm capable of mapping out the Pareto frontier in one pass (Section 2.1) and propose a modification that can be used to select robust clusterings with quantified uncertainties from the frontier when the sampling error is significant (Section 2.2). We then present our analytic and experimental results in Section 3. In Section 3.1, we provide evidence for the sparsity of the Pareto frontier, giving hope that it is possible to efficiently study it in practice. To demonstrate our algorithm and build intuition for the Pareto frontier structure, we apply it to three different DIB optimization problems in Section 3.2: compressing the English alphabet, extracting informative color classes from natural images, and discovering group–theoretical structure. Finally, in Section 4, we discuss related results and directions for future work.

2. Methods

We design our algorithm around two main requirements: firstly, we would like to be able to optimize the primal objective of Equation (1) directly, thereby allowing us to discover convex portions of the frontier; secondly, we would like a method that records the entire frontier in one pass rather than finding a single point with each run. While the task of finding the exact Pareto frontier is expected to be hard in general, Theorem A1 applied to Example 1, gives us hope that the size of the Pareto frontier grows at most polynomially with input size $|X|$. As is often the case when dealing with the statistics of extreme values, we expect that points near the frontier are rare and propose a pruned search technique with the hope that significant portions of the search space can be pruned away in practice. In the spirit of the probabilistic analysis provided above, we would like an algorithm that samples from a distribution that favors near-optimal encoders, thereby accelerating the convergence of our search. For this reason, we favor an agglomerative technique, with the intuition that there are good encoders that can be derived from merging the output classes of other good encoders. An agglomerative approach has the additional benefit of being able to record the entire frontier in one pass. For these reasons, we propose an agglomerative pruned search algorithm for mapping the Pareto frontier in Section 2.1. We also describe in Section 2.2 a modification of the algorithm that can be applied to situations where only a finite number of samples are available.

2.1. The Pareto Mapper

Our method, dubbed the *Pareto Mapper* (Algorithm 1), is a stochastic pruned agglomerative search algorithm with a tunable parameter ϵ that controls the search depth. The algorithm is initialized by enqueuing the identity encoder, $f = \mathrm{id}$, into the search queue. At each subsequent step (illustrated in Figure 2), an encoder is dequeued. A set containing all of the Pareto-optimal encoders thus far encountered is maintained as the algorithm proceeds. All encoders that can be constructed by merging two of the given encoder's output classes (there are $O(n^2)$ of these) are evaluated against the frontier of encoders seen so far; we call encoders derived this way child encoders. If a child encoder is a distance d from the current frontier, we enqueue its children with probability $e^{-d/\epsilon}$ and discard it otherwise, resulting in a search over an effective length-scale ϵ from the frontier. The selection of ϵ tunes the trade-off between accuracy and search time: $\epsilon = 0$ corresponds to a greedy search that does not evaluate sub-optimal encodings, and $\epsilon \to \infty$ corresponds to a brute-force search over all possible encodings. As the search progresses, the Pareto frontier is refined, and we are able to prune a larger majority of the proposed encoders. The output of our algorithm is a Pareto set of all found Pareto optimal clusterings of the given trade-off.

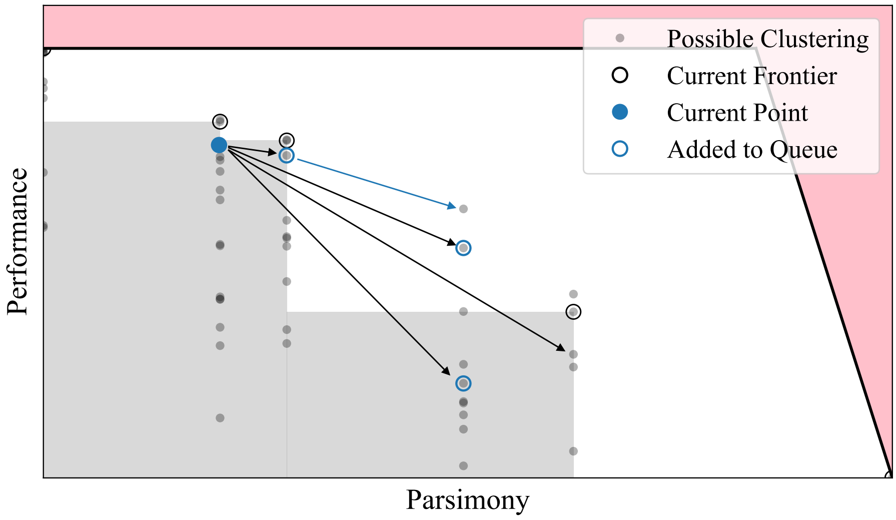
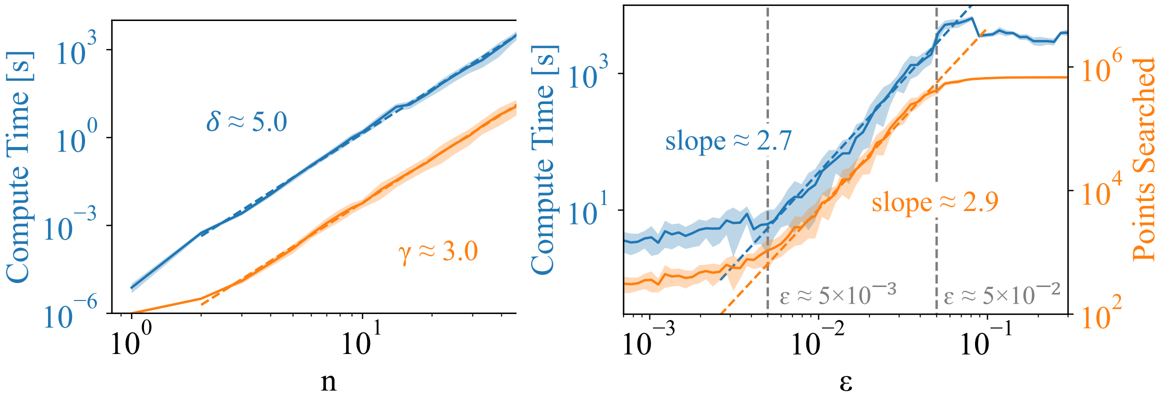


Figure 2. Illustration of one step in the main loop of the Pareto Mapper (Algorithm 1). For pedagogical purposes, all possible encoders (filled gray circle) are plotted on the objective plane. The Pareto optimal points searched so far are marked with open black circles, and the region of the objective plane they dominate is shaded in gray. The black arrows show neighboring encoders and newly enqueued encoders are marked by open blue circles; encodings that are optimal with respect to the current frontier are enqueued with certainty and sub-optimal encodings enqueued with probability $e^{-d/\epsilon}$, where d is the distance from the frontier. Note that some Pareto optimal points are only accessible through sub-optimal encoders (blue arrow).

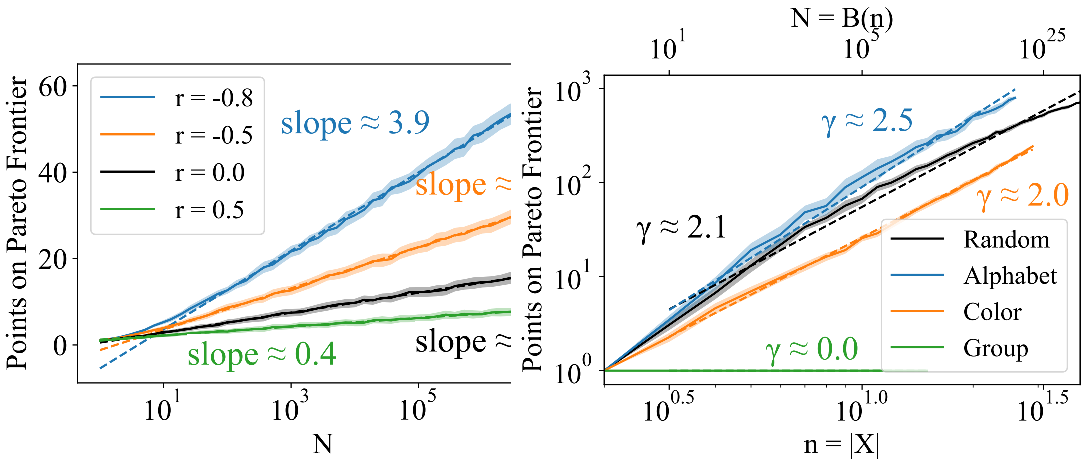
The Pareto frontier at any given moment is stored in a data structure, called a *Pareto Set*, which is a list of point structures. A point structure, p, contains fields for both objectives p.x, p.y, and optional fields for storing the uncertainties p.dx, p.dy and encoding function p.f. The Pareto Set is maintained so that the Pareto-optimality of a point can be checked against a Pareto Set of size m in $\Theta(\log m)$ operations. Insertion into the data structure requires in the worst case $\Theta(m)$ operations, which is optimal, as a new point could dominate $\Theta(m)$ existing points necessitating their removal. We define the distance from the frontier as the minimum Euclidean distance that a point would need to be displaced before it is Pareto-optimal, which also requires in the worst case $\Theta(m)$ operations. A list of pairs $(H(Z), I(Z;Y))$, sorted by its first index, provides a simple implementation of the Pareto Set. The pseudocode for important auxiliary functions such as PARETO_ADD and PARETO_DISTANCE is provided in Appendix B.

Although we have provided evidence for polynomial scaling of size of the Pareto set, it is not obvious if the polynomial scaling of the Pareto set translates to the polynomial scaling of our algorithm, which depends primarily on how quickly the search space can be pruned away by evaluation against the Pareto frontier. To demonstrate the polynomial scaling of our algorithm with n, we evaluate the performance of the Pareto Mapper on randomly generated p_{XY}. Since $\epsilon \to \infty$ corresponds to a brute-force search, and therefore has no hope of having polynomial runtime, we focus on the $\epsilon \to 0$ case; we show later, in Section 3.2.1, that $\epsilon \to 0$ is often sufficient to achieve good results. For Figure 3a, we randomly sample p_{XY} uniformly over the simplex of dimension $|X||Y| - 1$ varying $|X|$ with fixed $|Y| = 30$. We find that the scaling is indeed polynomial. Comparing with the scaling of the size of the Pareto set shown in Figure 4b, we see that approximately $O(n)$ points are searched for each point on the Pareto frontier. While the computation time, empirically estimated to be $\Theta(n^{5.0})$, is limiting in practice, we note that it is indeed polynomial, which is sufficient for our purposes.



(**a**) (**b**)

Figure 3. Scaling of computation time (**left** scale) and points searched (**right** scale) as a function of (**a**) input size n at $\epsilon = 0$, where we find compute time scales as $O(n^\delta)$ and the size of the Pareto set scales as $O(n^\gamma)$; (**b**) search parameter ϵ for randomly generated p_{XY} of fixed size.



(**a**) (**b**)

Figure 4. Scaling of number of points on the Pareto frontier (**a**) as a function of $N = |S|$ for bivariate Gaussian distributed (U, V) with specified correlation $r \equiv \sigma_{UV}/\sigma_U\sigma_V$, and (**b**) for the DIB problem with input size n where we find $|\,\text{Pareto}(S)| = O(n^\gamma)$.

We also evaluate the scaling of our algorithm with ϵ. Again, we randomly sample p_{XY} uniformly over the simplex of dimension $|X||Y| - 1$, this time fixing $|X| = 11$ and $|Y| = 30$, with results plotted in Figure 3b. We find that the relevant scale for distances is between $\epsilon_- \approx 5 \times 10^{-3}$ and $\epsilon_+ \approx 5 \times 10^{-2}$; while the specifics of the characteristic range for ϵ depends on the dataset, we empirically find that while ϵ_+ remains constant, ϵ_- decreases as n increases. This is consistent with the fact that as n increases, the DIB plane becomes denser, and the average separation between points decreases. This would suggest that there is an n above which the runtime of the Pareto Mapper exhibits exponential scaling for any $\epsilon > 0$. In the absence of noise, one can run the Pareto Mapper at a number of different values of ϵ evaluating precision and recall with respect to the combined frontier to evaluate convergence (see Figure 8b). We discuss how to set ϵ in the presence of noise due to sampling in Section 3.2.1.

2.2. Robust Pareto Mapper

So far, we have assumed access to the true joint distribution p_{XY}. Normally, in practice, we are only provided samples from this distribution and must estimate both objective functions, the mutual information $I(Z;Y)$ and entropy $H(Z)$, from the empirical distribution $\hat{p}_{XY}$. Despite the uncertainty in these estimates, we would like to find clusterings that are Pareto optimal with respect to the true distribution. Here, we propose a number of modifications to Algorithm 1 that allow us to quantify our uncertainty about the frontier and thereby produce a frontier that is robust to sampling noise. The modified algorithm, dubbed the *Robust Pareto Mapper* (Algorithm 2), is described below.

Given samples from the joint distribution, we construct the empirical joint distribution and run the Pareto Mapper (Algorithm 1) replacing the entropy and mutual information functions with their respective estimators. We use the entropy estimator due to Nemenman, Shafee, and Bialek (NSB) [21], as it is a Bayesian estimator that provides not only a point estimate but also provides some bearing on its uncertainty, although another suitable estimator can be substituted. We find that our method works well even with point estimators, in which case resampling techniques (e.g., bootstrapping) are used to obtain the uncertainty in the estimate. After running the Pareto Mapper, points that are not significantly different from other points in the Pareto set are removed. This filtering operation considers points in order of ascending uncertainty, as measured by the product of its standard deviations in $H(Z)$, and $I(Z;Y)$. Subsequent points are added as long as they do not overlap with the confidence interval in either H or I with a previously added point, and they are removed otherwise. There is some discretion in choosing the confidence interval, which we have chosen empirically to keep the discovered frontier robust between independent runs. This filtering step is demonstrated in Section 3.2.1.

Algorithm 2 Robust Pareto Mapper: dealing with finite data

Input: Empirical joint distribution $\hat{p}_{XY}$, search parameter ε, and sample size S
Output: Approximate Pareto frontier P with uncertainties

```
procedure ROBUST_PARETO_MAPPER($\hat{p}_{XY}, \varepsilon$)
    Pareto Set $P \leftarrow$ PARETO_MAPPER($\hat{p}_{XY}, \epsilon$)    ▷ Run PARETO_MAPPER with suitable estimators
    Pareto Set $P' = \emptyset$    ▷ Initialize set of robust encoders
    for $p \in P$ do    ▷ This step can be skipped if an interval estimator is used above
        ($p$.dx, $p$.dy) $\leftarrow$ RESAMPLE($p, \hat{p}_{XY}$)    ▷ Store uncertainty of points on frontier
    for $p \in P$ in ascending order of uncertainty do
        if $p$ is significantly different than all $q \in P'$ then
            $P' \leftarrow$ PARETO_ADD($p, P'$)    ▷ Filter with preference for points with low variance
    return $P'$
```

3. Results

3.1. General Properties of Pareto Frontiers

Before introducing the specifics of the DIB problem, we would like to understand a few general properties of the Pareto frontier. The most immediate challenge we face is the size of our search space. For an input of size $|X|$, the number of points on the DMC frontier is bounded by $|X|$, but there is no such limit on the DIB frontier. Given the combinatorial growth of the number of possible clusterings with $|X|$, it is not immediately clear that it is possible to list all of the points on the frontier, let alone find them. If we are to have any chance at discovering and specifying the DIB frontier, it must be that DIB-optimal points are sparse within the space of possible clusterings, where sparse is taken to mean that the size of the frontier scales at most polynomially with the number of items to be compressed.

In this section, we provide sufficient conditions for the sparsity of the Pareto set in general and present a number of illustrative examples. We then apply these scaling relationships to the DIB search space and provide numerical evidence that the number of points grows only polynomially with $n \equiv |X|$ for most two-objective trade-off tasks.

3.1.1. Argument for the Sparsity of the Pareto Frontier

First, we will formally define a few useful terms. Let $S = \{\vec{s}_i\}_{i=1}^N$ be a sample of N independent and identically distributed (i.i.d.) bivariate random variables representing points $\vec{s}_i = (U_i, V_i)$ in the Pareto plane.

Definition 1. *A point $(U, V) \in S$ is maximal with respect to S, or equivalently called Pareto-optimal, if $\forall (U_i, V_i) \in S$, $V_i > V \implies U_i > U$. In other words, a point is maximal with respect to S if there are no points in S both above and to its left (in our picture with the horizontal axis flipped).*

Definition 2. *For a set of points $S \subset \mathbb{R}^2$, the Pareto set* $\text{Pareto}(S) \subseteq S$ *is the largest subset of S such that all $(U, V) \in \text{Pareto}(S)$ are maximal with respect to S.*

Now, we can state the main theorem of this section, which we prove in Appendix A.

Theorem 1. *Let $S = \{(U_i, V_i)\}_{i=1}^N$ be a set of bivariate random variables drawn i.i.d. from a distribution with Lipschitz continuous CDF $F(u, v)$, and invertible marginal CDFs F_U, F_V. Define the region*

$$R_N \equiv \left\{ (u, v) \in [0,1] \times [0,1] : u + v - C(u, v) \geq e^{-\frac{1}{N}} \right\} \tag{3}$$

where $C(u,v)$ *denotes the copula of* (U_i, V_i)*, which is the function that satisfies* $F(u,v) = C(F_U(u), F_V(v))$*.*

Then, if the Lebesgue measure of this region $\lambda(R_N) = \Theta\left(\frac{\ell(N)}{N}\right)$ *as* $N \to \infty$*, we have* $\mathbb{E}\big[|\operatorname{Pareto}(S)|\big] = \Theta(\ell(N))$*.*

Example 1. *Let us consider the case of independent random variables with copula* $C(u,v) = uv$*. Note that in this case, the level curves* $u + v - C(u,v) = e^{-\frac{1}{N}}$ *are given by* $v = \frac{e^{-\frac{1}{N}} - u}{1-u}$*. We can then integrate to find the area of the region* R_N

$$\lambda(R_N) = 1 - \int_0^{e^{-\frac{1}{N}}} \frac{e^{-\frac{1}{N}} - u}{1-u} du = e^{-1/n}\left(1 - e^{1/n}\right)\left(\log\left(1 - e^{-1/n}\right) - 1\right) \qquad (4)$$

Expanding for large N*, we find that* $\lambda(R_N) = \frac{\log N}{N} + O(N^{-1})$*. We see that this satisfies the conditions for Theorem A1 with* $\ell(N) = \log N$*, giving* $\mathbb{E}_S\big[|\operatorname{Pareto}(S)|\big] = \Theta(\log N)$*.*

Additional examples can be found in Appendix A. Numerically, we see that for independent random variables U and V, the predicted scaling holds even down to relatively small N; furthermore, the linear relationship also holds for correlated Gaussian random variables U, V (Figure 4a). The logarithmic sparsity of the Pareto frontier allows us to remain hopeful that it is possible, at least in principle, to fully map out the DIB frontier for deterministic encodings of discrete domains despite the super-exponential number of possible encoders.

3.1.2. Dependence on Number of Items $|X|$

The analysis above gives us hope that Pareto-optimal points are generally polylogarithmically sparse in $N \equiv |S|$, i.e., $|\operatorname{Pareto}(S)| = O((\log N)^\gamma)$. Of course, the scenario with which we are concerned is one where the random variables $U = -H(Z)$ and $V = I(Z;Y)$; the randomness of U and V in this case comes from the choice of encoder $f : X \to Z$ which, for convenience, we assume is drawn i.i.d. from some distribution over the space of possible encodings. Note that conditioned on the distribution of X and Y, the points $(H(f(X)), I(f(X);Y))$ are indeed independent, although the agglomerative method we use to sample from the search space introduces dependence; however, in our case, this dependence likely helps the convergence of the algorithm.

In the DIB problem, and clustering problems more generally, we define $n \equiv |X|$ to be the size of the input. The search space is over all possible encoders $f : X \to Z$, which has size $N = B(n)$ where $B(n)$ are the Bell numbers. Asymptotically, the Bell numbers grow super-exponentially: $\ln B(n) \sim n \ln n - n \ln \ln n$, making an exhaustive search for the frontier intractable. We provide numerical evidence in Figure 4b that the sparsity of the Pareto set holds in this case, with its size scaling as $O(\operatorname{poly}(n))$, or equivalently, $O(\operatorname{polylog}(N))$. While in the worst case, all B_n clusterings can be DIB-optimal (the case where $(p_{XY})_{ij} = \operatorname{diag}(\vec{r})$ for r_i drawn randomly from the $(n-1)$-dimensional simplex results in clusterings with strict negative monotonic dependence on the DIB plane, and therefore all points are DIB-optimal, see Appendix A), our experiments show that in practice, the size of the Pareto frontier (and compute time) grows polynomially with the number of input classes n (Figure 3), with the degree of the polynomial depending on the details of the joint distribution $p(x,y)$. This result opens up the possibility of developing a tractable heuristic algorithm that maps out the Pareto frontier, which we will explore in the remainder of this paper.

3.2. At the Pareto Frontier: Three Vignettes

The purpose of this section is to demonstrate our algorithm and illustrate how the primal DIB frontier can be used for model selection and to provide additional insights about the data. To this end, we apply our algorithm to three different DIB optimization tasks: predicting each character from its predecessor in the English text, predicting an object from its color, and predicting the output of a group multiplication given its input. In all cases, the goal is to find a representation that retains as much predictive power as possible given a constraint on its entropy. We will describe the creation of each dataset and motivate its study. For all three tasks, we discuss the frontier discovered by our algorithm and highlight informative points on it, many of which would be missed by other methods either because they are not DMC-optimal or because they lie within convex regions of the frontier. For the task of predicting the subsequent English character, we will also compare our algorithm to existing methods including the Blahut–Arimoto algorithm, and a number of geometric clustering algorithms; we will also use this example to demonstrate the Robust Pareto Mapper (Algorithm 2).

3.2.1. Compressing the English Alphabet

First, we consider the problem of compressing the English alphabet with the goal of preserving the information that a character provides about the subsequent character. In this case, we collect 2-gram statistics from a large body of English text. Treating each character as a random variable, our goal is to map each English character X into a compressed version Z while retaining as much information as possible about the following character Y.

Our input dataset is a 27 × 27 matrix of bigram frequencies for the letters a–z and the space character, which we denote "_" in the figures below. We computed this matrix from the 100 Mb *enwiki8* (http://prize.hutter1.net/ (accessed on 15 February 2020)) Wikipedia corpus after removing all symbols except letters and the space character, removing diacritics, and making all letters lower-case.

The Pareto frontier is plotted in Figure 5 and the points corresponding to some interesting clusterings on the frontier are highlighted. We find that from 2-gram frequencies alone, the DIB-optimal encodings naturally discover relevant linguistic features of the characters, such as their grouping into vowels and consonants.

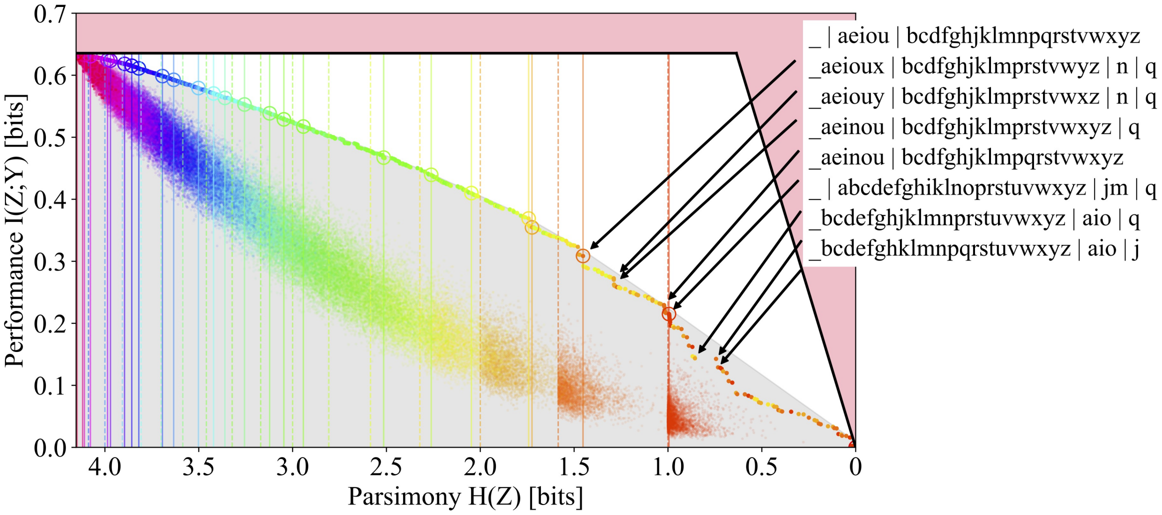


Figure 5. The primal DIB frontier of the English alphabet is compressed to retain information about the subsequent character. Points in the shaded gray region, indicating the convex hull, are missed by optimization of the DIB Lagrangian. Encodings corresponding to interesting features of the frontier are annotated, and DMC-optimal points are circled. Dotted vertical lines mark the location of balanced clusters (i.e., $H(Z) = \log_2 k$ for $k \in \{2, 3, \ldots\}$), and solid vertical lines correspond to the entropy of the DMC-optimal encodings. A sample of encodings drawn uniformly at random for each $|Z|$ is evaluated on the plane, illustrating the large distance from the frontier for typical points in the search space. The color indicates $|Z|$.

The DMC-optimal encoding corresponding to a cluster size of $k = 2$ is nearly balanced (i.e., $H(Z) = \log_2 2$) and separates the space character and the vowels from most of the consonants. However, in contrast to the binary classification case of $|Y| = 2$ studied in [13], the DMC-optimal encodings are far from balanced (i.e., $H(Z) \approx \log_2(k)$) for larger k. We note that on the DIB frontier, the most prominent corners are often not the DMC-optimal points, which are circled. By looking at the corners and the DMC-optimal points near the corners, which are annotated on the figure, we discover the reason for this: distinguishing anomalous letters such as 'q' has an outsized effect on the overall information relative to its entropy cost. These features are missed when looking only at DMC-optimal points, because although the 2-gram statistics of 'q' are quite distinct (it is almost always followed by a 'u'), it does not occur frequently enough to warrant its own cluster when our constraint is cluster count rather than entropy. In other words, 'q' is quite special and noteworthy, and our Pareto plot reveals this while the traditional DMC or DIB plots do not.

The frontier is seen to reveal features at multiple scales, the most prominent corner corresponding to the encoding that separates the space character, '_', from the rest of the alphabet, and the separation of the vowels from (most of the) consonants. The separation of 'q' often results in a corner of a smaller scale because it is so infrequent. These corners indicate the natural scales for hierarchical clustering. We also note that a large majority of the points, including those highlighted above, are below the convex hull (denoted by the solid black line) and are therefore missed by algorithms that optimize the DIB Lagrangian.

A random sample of clusterings colored by $|Z|$ is also plotted on the DIB plane in Figure 5; a sample for each value of $|Z| = \{1, \ldots, |X|\}$ is selected uniformly at random. We see that there is a large separation between the typical clustering, sampled uniformly at random, and the Pareto frontier, indicating that a pruned search based on the distance from the frontier, such as the Pareto Mapper of Algorithm 1, is likely able to successfully prune away much of the search space. A better theoretical understanding of the density could provide further insights on how the runtime scales with ϵ.

We now compare the results of the Pareto Mapper (Algorithm 1) with other clustering methods. We first use the Pareto Mapper with $\epsilon = 0$ to derive a new dataset from the original alphabet dataset (which has $|X| = 27$) by taking the $|Z| = 10$ clustering with the highest mutual information. We are able to obtain a ground truth for this new dataset with $|X| = 10$ using a brute force search, against which we compare the other methods. These methods are compared on the DIB plane in Figure 6 and in tabular form in Table 1. Notably, all the encodings found by the Blahut–Arimoto algorithm used in [2,3] are DIB-optimal, but as it optimizes the DIB Lagrangian, it is unable to discover the convex portions of the frontier. We also compare our algorithm to geometric clustering methods where we assign clusters pairwise distances according to the Jensen–Shannon distances between the conditional distributions $p(Y|X = x_i)$. These methods perform poorly when compared on the DIB plane for a number of reasons: firstly, some information is lost in translation to a geometric clustering problem, since only pairwise distances are retained; secondly, the clustering algorithms are focused on minimizing the number of clusters and are therefore unable to find more than n points. Additionally, these geometric clustering algorithms, while similar in spirit, are not directly optimizing the DIB objective.

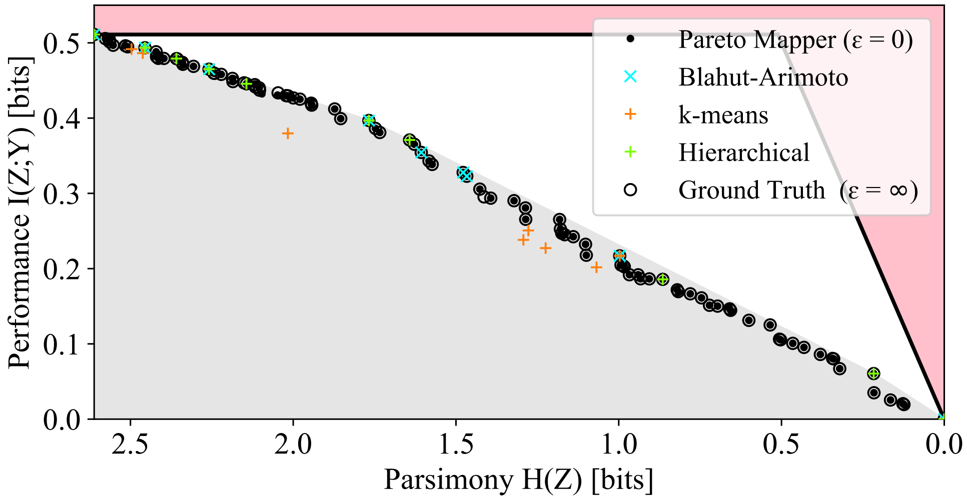
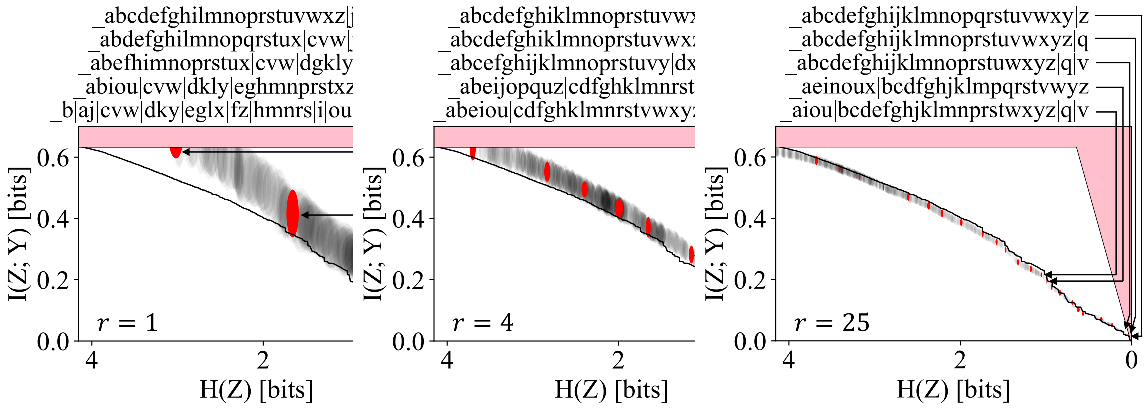


Figure 6. Comparison of the Pareto Mapper and other classification algorithms with ground truth for $|X| = 10$. The true Pareto frontier is calculated with a brute force search over all $B(10) = 115,975$ clusterings f.

Table 1. Comparison of the performance of Algorithm 1 with other clustering algorithms. Here, a true positive (TP) is a point that is correctly identified as being Pareto optimal by a given method; false positives (FP) and false negatives (FN) are defined analogously.

Method	**Points**	**TP**	**FP**	**FN**	**Precision**	**Recall**
Ground truth ($\epsilon \to \infty$)	94	94	0	0	1.00	1.00
Pareto Mapper ($\epsilon = 10^{-2}$)	94	94	0	0	1.00	1.00
Pareto Mapper ($\epsilon = 0$)	91	88	3	6	0.97	0.94
Hierarchical (average)	10	7	3	87	0.70	0.07
Hierarchical (single)	10	10	0	84	1.00	0.11
Hierarchical (Ward)	10	7	3	87	0.70	0.07
k-means (JSD)	10	3	7	91	0.30	0.02
k-means (wJSD)	10	2	8	92	0.20	0.10
Blahut Arimoto	9	9	0	85	1.00	0.10

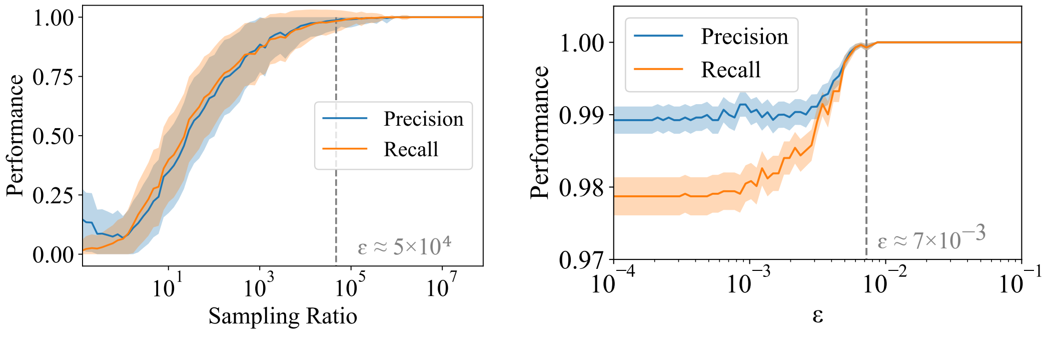
To demonstrate the Robust Pareto Mapper (Algorithm 2), we create a finite sample $\hat{n}_{XY} = s\hat{p}_{XY}$ from a multinomial distribution with parameter p_{XY} and s trials. To quantify the sample size in natural terms, we define the sampling ratio $r \equiv s/2^{H(X,Y)}$. The results of the Robust Pareto Mapper on the alphabet dataset for several sampling ratios are shown in Figure 7. We note that even for relatively low sampling ratios, the algorithm is able to extract interesting information; it is able to quickly separate statistically distinct letters such as 'q' and identify groups of characters such as vowels. As the sampling ratio increases, the Robust Pareto Mapper identifies a larger number of statistically significant clusterings (marked in red) from the rest of the discovered frontier (marked in gray). It is also notable that uncertainties in the entropy are typically lowest for encodings that split X into roughly equally probably classes; that these clusters are preferred is most readily seen in the highlighted clustering with $H(Z) \approx 1$ in Figure 7b. We can see from these plots that, especially for low sampling ratios, the estimated frontier often lies above that of the true p_{XY} (solid black line). This is expected, as estimators for mutual information are often biased high. Despite this, the true frontier is found to lie within our estimates when the variance of the estimators is taken into account even for modest sampling ratios, as seen in the plot for $r = 4$.



(**a**) (**b**) (**c**)

Figure 7. The optimal frontier discovered by the Robust Pareto Mapper at various sampling ratios. The points corresponding to robust clusterings selected by the algorithm are highlighted in red, with the rest in gray. The true frontier is shown in solid black.

Finally, we would like to comment on choosing the parameter ϵ in Algorithms 1 and 2 when working with limited sample sizes. The uncertainty in the frontier due to finite sampling effects naturally sets a scale for choosing ϵ. Ideally, we want the two length scales—that given by ϵ, and that due to the variance in the estimators—to be comparable. This ensures that we are not wasting resources fitting sampling noise. Evaluating the performance as a function of sample size and epsilon, we see that often, sample size is the limiting factor even up to significant sampling ratios, and often, a small ϵ is often sufficient. This is demonstrated in Figure 8, where it can be seen that performance is good even with small ϵ, and increasing ϵ does not result in a more accurate frontier until the sampling ratio is greater than $r \approx 5 \times 10^4$. In practice, determining the appropriate ϵ can be accomplished by selecting different holdout sets, and running the algorithm at a given ϵ in each case; when ϵ is chosen appropriately, the resulting Pareto frontier should not vary significantly between the runs.



(**a**) (**b**)

Figure 8. Performance as a function of (**a**) sample size, and (**b**) ϵ. The precision and recall are measured relative to the true frontier obtained by a brute force search on the true distribution.

3.2.2. Naming the Colors of the Rainbow

Human languages often have a small set of colors to which words are assigned, and they remarkably often settle on similar linguistic partitions of the color space despite cultural and geographic boundaries [26]. As our next example, we apply our method to the problem of optimally allocating words to colors based on the statistics of natural images. In order to cast this as a DIB-style learning problem, we consider the goal of being able to identify objects in natural images based solely on color: the variable we would like to predict, Y, is therefore the class of the object (e.g., apple or banana). The variable we would like to compress, X, is the average color of the object. The Pareto-optimal classifiers are those that, allotted limited memory for colorative adjectives, optimally draw the boundaries to accomplish the task of identifying objects. We demonstrate some success in discovering different color classes, relate it to those typically found in natural languages, and discuss shortcomings of our method.

We create a dataset derived from the COCO dataset [27], which provides a corpus of segmented natural images with 91 object classes. There are a number of challenges we immediately face in the creation of this dataset, which require us to undertake a number of preprocessing steps. Firstly, using standard RGB color values, with 8 bits per channel, leaves over 16 million color classes to cluster, which is not feasible using our technique. Secondly, RGB values contain information that is not relevant to the task at hand, as they vary with lighting and image exposure. Thus, we turn to the HSV color model and use only the hue value (since hue is a circular quantity, we use circular statistics when discussing means and variances), which we refer to as the color of the object from now on. This leaves 256 values which are further reduced by contiguous binning so that each bin has roughly equal probability in order to maximize the entropy of X. After this preprocessing, we are left with an input of size $|X| = 30$. Another challenge we face is that there are often cues in addition to average color when performing object identification such as color variations, shape, or contextual understanding of the scene; in order to obtain the cleanest results, we retain only those classes that could reasonably be identified by color alone. Specifically, for the roughly 800,000 image segments from the approximately 100,000 images we considered in the COCO dataset, we calculate the average color of each segment and keep only the 40% with the most uniform color as measured by the variance of the hue across the segment; then, looking across classes, we keep only those that are relatively uniform on the average color of its instances, keeping approximately the most uniform 20% of classes. We are left with a dataset of approximately 80,000 objects across $|Y| = 18$ classes, predictably including rather uniformly colored classes such as apples, bananas, and oranges. We chose these cutoff percentiles heuristically to maximize the predictive power of our dataset while maintaining a sizable number of examples.

The Pareto frontier for this dataset is shown in Figure 9. A number of DMC-optimal points are circled, and their respective color palettes are plotted below in descending order of likelihood. First, we note that the overall amount of relevant information is quite low, with a maximum $I(X;Y) \approx 0.12$, indicating that despite our preprocessing efforts, color is not a strong predictor of object class. Unlike the other Pareto frontiers considered, there are a few prominent corners in this frontier, which is a sign that there is no clear number of colors to best resolve the spectrum. For the first few DMC-optimal clusterings, the colors fall broadly into reddish-purples, greens, and blues. This is somewhat consistent with the observation that languages with limited major color terms often settle for one describing warm colors and one describing colder colors [26].

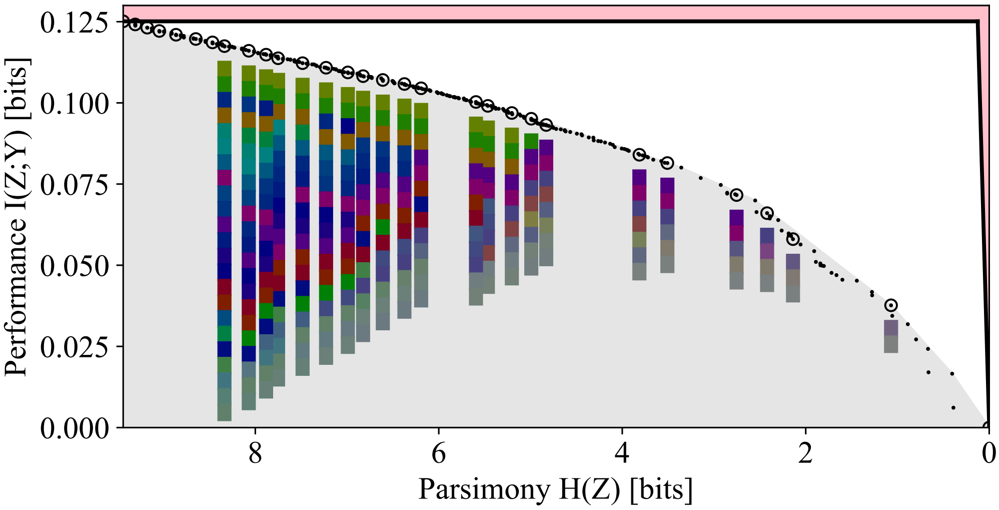
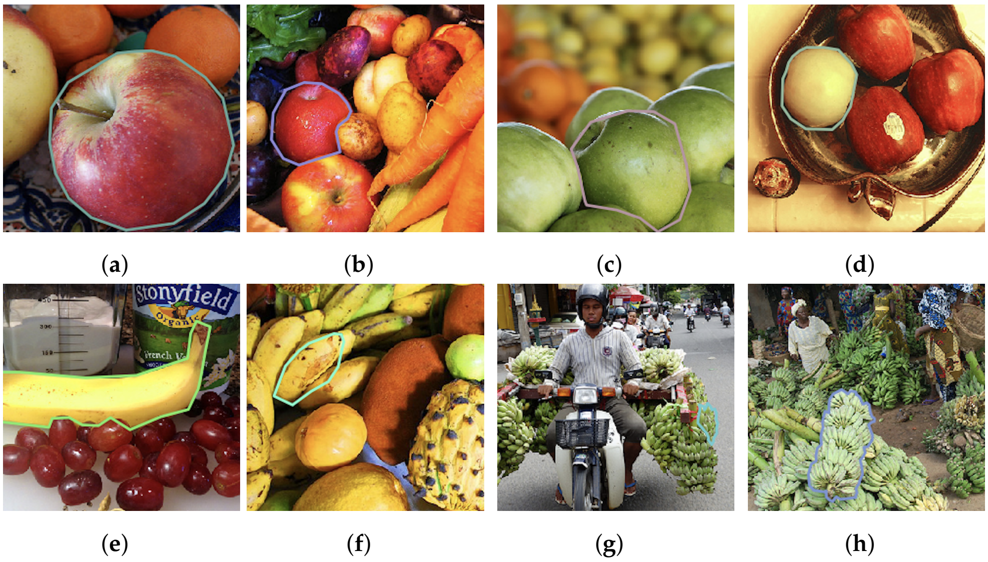


Figure 9. Pareto frontier of color data. A representative color patch for each cluster is shown below select points sorted by likelihood. The saturation of the patch represents the likelihood-weighted variance of the colors mapped to the class.

Overall, the results are not conclusive. We will address a few issues with our method and discuss how it might be improved. Firstly, as noted by [26], the colors present in human languages often reflect a communicative need and therefore should be expected to depend strongly on both the statistics of the images considered and also the prediction task at hand. Since the COCO dataset was not designed for the purpose of learning colors, classes had color outliers, despite our preprocessing efforts, which reduced the classification accuracy by color alone. Using color as a predictor of the variety of an apple or as a predictor of the ripeness of a banana might yield better results (see Figure 10); indeed, these tasks might be more reflective of the communicative requirements under which some human languages developed [26]. Due to the scarcity of relevant datasets, we have not attempted to address these subtleties.


(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 10. Examples of correctly (**a**,**b**) and incorrectly (**c**,**d**) identified apples; and correctly (**e**,**f**) and incorrectly (**g**,**h**) identified bananas from the filtered COCO dataset using the best discovered five-bin clustering.

Another issue, more fundamental to the DIB algorithm, is that DIB is not well suited for the compression of domains of a continuous nature. The DIB trade-off naturally favors a discrete domain, X, without a measure of similarity between objects in X. Unlike the other examples considered, the space of colors is inherently continuous: there is some notion of similarity between different hues. One weaknesses of the DIB trade-off is that it does not respect this natural notion of closeness and it is as likely to map distant hues together as it is ones that are close together. This is undesirable in the case of the color dataset, as we would ideally like to map contiguous portions of the color space to the same output. Other objectives, such as the IB or a multidimensional generalization of [13], may be more suitable in cases where the domain is of a continuous nature.

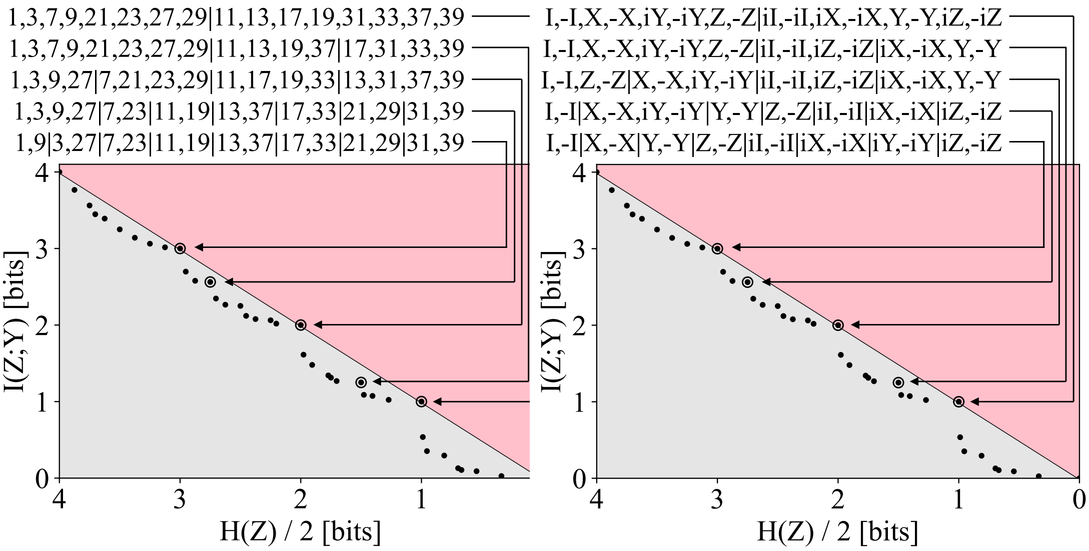
3.2.3. Symmetric Compression of Groups

For our final example, we turn our attention to a group–theoretic toy example illustrating a variation on the compression algorithm so far considered which we call "symmetric compression." We consider a triplet of random variables (X_1, X_2, Y), each taking on values in the set G with the special property that G forms a group under the binary group operation '$\cdot$'. We could apply Algorithm 1 directly to this problem by setting $X = (X_1, X_2)$, but this is not ideal, as it does not make use of the structure we know the data to have and as a result needlessly expands our search space. Instead, we make the slight modification, detailed in Appendix C, where we apply the same clustering to both inputs, $Z = (f(A), f(B))$. We would like to discover an encoding f that trades off the entropy of the encoding with the ability to predict Y from $(f(X_1), f(X_2))$. We expect that the DIB frontier encodes information about the subgroups of the group G, but we also expect to find points on the frontier corresponding to near-subgroups of G.

We consider two distributions. The first consists of the sixteen integers that are coprime to 40, i.e., $\{1,3,7,9,11,13,17,19,21,23,27,29,31,33,37,39\}$, which for a multiplicative group modulo 40 denoted $(\mathbb{Z}/40\mathbb{Z})^{\times}$. The second is the Pauli group G_1, whose elements are the sixteen 2×2 matrices generated by the Pauli matrices under matrix multiplication: they are the identity matrix I and Pauli matrices X, Y, Z, each multiplied by ± 1 and $\pm i$. These groups are chosen as they both have order 16 but are otherwise quite different; notably, $(\mathbb{Z}/40\mathbb{Z})^{\times}$ is abelian while G_1 is not. The joint probability distribution is defined as follows for each group G: we take (X_1, X_2) to be distributed uniformly over G^2 and $Y = X_1 \cdot X_2$. The distribution $p_{X_1 X_2 Y}$ is given as input to the symmetric Pareto Mapper (Algorithm A4).

The resultant frontiers are shown in Figure 11. As expected, the subgroups are readily identified in both cases, as seen the in circled points on the frontier with entropy $H(Z) = 1$, $H(Z) = 2$, and $H(Z) = 3$, corresponding to subgroups of size 2, 4, and 8, respectively. In this example, we also see that the clusterings corresponding to the subgroups saturate the feasibility bound of $I(Z;Y) = H(Z)$, indicating that at these points, all the information captured in Z is relevant to Y. At these points, the encoding effectively identifies a subgroup $H \leq G$ and retains information only about which of the $|G|/|H|$ cosets an element belongs to; as it retains the identity of the cosets of X_1 and X_2 in Z_1 and Z_2, it is able to identify the coset of the output Y, thereby specifying Y to $\log_2 \frac{|G|}{|H|}$ bits. These clearly desirable solutions show up prominently in the primal DIB frontier, yet their prominence is not evident on the frontier of the Lagrangian DIB—notably having zero kink angle as defined by [18].

In addition to the points corresponding to identified subgroups, a number of intermediary points have also been highlighted showing 'near-subgroups', where, allotted a slightly larger entropy budget, the encoder can further split cosets apart in such a way that partial information is retained. Interestingly, despite being very different groups, they have identical Pareto frontiers. This is because they both have subgroups of the same cardinality, and the entropy and relevant information of these encodings is agnostic to the group theoretic details and concerns itself only with the ability to predict the result of the group operation.



(**a**) (**b**)

Figure 11. Discovered frontier of the (**a**) $(\mathbb{Z}/40\mathbb{Z})^{\times}$ group and (**b**) the non-abelian Pauli group. Both groups have identical frontiers despite having different group structures.

4. Discussion

We have presented the Pareto Mapper algorithm, which computes the optimal trade-off between parsimony and performance for lossy data compression. By applying it to examples involving linguistics, image colors and group theory, we have demonstrated the richness of the DIB Pareto frontier that customarily lies hidden beneath the convex hull. Our English alphabet example revealed features at multiple scales and examples of what the frontier structure reveals about the data, and we demonstrated a modification to our algorithm that can aid model selection given significant sampling noise. Notably, we showed how the prominence of a point on the primal frontier can be a sharper tool for model selection than existing measures on the Lagrangian DIB frontier; for example, for our group theory examples, it outperformed the kink-angle method for model selection, which only gave kink angles of zero. Our datasets and implementation of the presented methods are freely available on GitHub (https://github.com/andrewktan/pareto_dib (accessed on 15 April 2022)).

Our result helps shed light on recently observed phases transitions. Recent work has shown that learning phase transitions can occur when optimizing two-objective trade-offs including the (D)IB [28–31] and β-VAEs [32]. In these cases, it is found that the performance of the learned representation makes discontinuous jumps as the trade-off parameter β is varied continuously. Such phase transitions can be readily understood in terms of the primal Pareto frontier of the trade-off: methods that optimize the Lagrangian DIB are only able to capture solutions on the convex hull of objective plane; as the Pareto frontier is largely convex, methods that optimize the Lagrangian exhibit will discontinuous jumps when the trade-off parameter β (which corresponds to the slope of a tangent to the frontier) is varied. This is analogous to the way first-order phase transitions in statistical physics arise, where it is the closely related Legendre–Fenchel dual that is minimized.

We would like to emphasize that, going beyond the IB framework, our basic method (Section 2.1) is generally applicable to a large class of two-objective optimization problems, including general clustering problems. Specifically, our method can be adapted for two-objective trade-offs with the following properties: a discrete search space; a frontier that, for typical datasets, grows polynomially with the input size $|X|$; and a notion of relatedness between objects in the search space (e.g., for the DIB problem, new encodings can be derived from existing ones by merging its output classes), which allows for an agglomerative search.

The modification (Section 2.2) can also be adapted given suitable estimators for other two-objective trade-offs.

Outlook

There are many opportunities to further improve our results both conceptually and practically. To overcome the limitations we highlighted with our image color dataset, it will be interesting to generalize our work and [13] to compressing continuous variables potentially with trade-offs such as the IB. While our evidence for the polynomial scaling of the size of the Pareto frontier is likewise applicable to other trade-offs of this sort, the runtime of our algorithm depends heavily on how quickly the search space can be pruned away and therefore is not guaranteed to be polynomial. Here, there is ample opportunity to tighten our analysis of the algorithmic complexity of finding the DIB frontier and on the scaling of generic Pareto frontiers.

Proofs aside, it will also be interesting to optimize the algorithm runtime beyond simply showing that it is polynomial. Although we have demonstrated the polynomial scaling of our algorithm for realistic datasets, the polynomial is of a high degree for our implementation, placing limits of $|X| \leq 50$ in practice. There are fundamental lower bounds on the runtime set by the scaling of the Pareto set, which we have shown in Figure 4b to be approximately $O(n^{2.1})$ for realistic datasets; however, there is likely to be some room for reducing the runtime by sampling clusterings from a better distribution. Another opportunity for improvement is increasing the speed at which a given point can be evaluated on the objective plane, which is evidenced by the gap between the runtime, approximately $O(n^{5.0})$, and the number of points searched, $O(n^{3.0})$ (Figure 3a).

While our method is only applicable to trade-offs over discrete search spaces, the Pareto frontier over continuous search spaces can also fail to be (strictly) concave. For example, the inability for the Lagrangian formulation of the (D)IB to explore all points on the trade-off has previously been studied in [12]. They propose a modification to the (D)IB Lagrangian that allows for the exploration of parts of the frontier that are not strictly concave. An interesting direction for future work is to study whether a similar modification to the Lagrangian can be used to discover the convex portions of similar trade-offs, including those over discrete spaces. Another direction for future work is to compare the primal DIB frontier with solutions to the IB; while solutions to the DIB Lagrangian often perform well on IB plane [3], it is an open question whether the solutions to the primal DIB perform favorably. Finally, as pointed out by a helpful reviewer, the dual problem corresponding to the primal problem of Equation (1), being a convex optimization problem, is also an interesting direction for future study.

We would also like to note that Pareto-pruned agglomerative search is a generic strategy for mapping the Pareto frontiers of two-objective optimization problems over discrete domains. The Pareto Mapper algorithm can also be extended to work in multi-objective settings given an appropriate implementation Pareto set in higher dimensions. We conjecture that the poly-logarithmic scaling of the Pareto set holds in higher dimensions as well. Extending this work to multi-objective optimization problems is another interesting direction for future work.

In summary, multi-objective optimization problems over discrete search spaces arise naturally in many fields from machine learning [3,10,18,19,33–35] to thermodynamics [36] and neuroscience [37]. There will therefore be a multitude of interesting use cases for further improved techniques that map these Pareto frontiers in full detail, including concave parts that reveal interesting structure in the data.

Author Contributions: Conceptualization, resources, supervision, project administration, funding acquisition, M.T. and I.L.C.; methodology, software, validation, formal analysis, investigation, writing—original draft preparation, writing—review and editing, visualization, A.K.T., M.T. and I.L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute for Artificial Intelligence and Fundamental Interactions (IAIFI) through NSF Grant No. PHY-2019786 and the Natural Sciences, the Engineering Research Council of Canada (NSERC) [PGSD3-545841-2020], the Casey and Family Foundation, the Foundational Questions Institute and the Rothberg Family Fund for Cognitive Science.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Ian Fischer for helpful discussions. We would also like to thank the anonymous reviewers for helpful clarifications and suggestions for future work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Proof of Pareto Set Scaling Theorem

As discussed in Section 3.1, the performance of our algorithm depends on the size of the Pareto frontier. In the paper, we provide experimental evidence for the polynomial scaling of the DIB Pareto frontier of a variety of datasets. In this appendix, we will prove Theorem A1, which provides sufficient conditions for the sparsity of the Pareto frontier and apply it to a number of examples.

As in Section 3.1, let $S = \{(U_i, V_i)\}_{i=1}^N$ be a sample of N i.i.d. bivariate random variables having joint cumulative distribution $F_{UV}(u, v)$. Further, let $R_{S,U}(U_i)$ and $R_{S,V}(V_i)$ be the marginal rank statistics of U and V, respectively, with respect to S; that is, U_i is the $R_{S,U}(U_i)^{\text{th}}$ smallest U-value in S and likewise for V. Ties can be broken arbitrarily. We will often drop the subscripts on $R_{S,U}$ and $R_{S,V}$ when it is clear by context.

Definition A1. *Given a permutation $\sigma : [N] \to [N]$ where $[N] \equiv \{1, \ldots, N\}$, we call i a sequential minimum if $j < i \Rightarrow \sigma(j) > \sigma(i)$.*

We would now like to show that the marginal rank statistics S are sufficient for determining membership in Pareto(S), which we formalize in Lemma A1.

Lemma A1. *Let $\sigma_U(i) = R(U_i)$ and $\sigma_V(i) = R(V_i)$. An element $(U_i, V_i) \in S$ is maximal if and only if its rank, i, is a sequential minimum of $\sigma_U \circ \sigma_V$.*

Proof. ($\Longrightarrow$) Assume $(U_i, V_{\sigma(i)}) \in S$ is maximal. For any other point $(u_j, v_{\sigma(j)}) \in S$, $i \neq j$, if $j < i \Rightarrow u_i < u_j$, then $v_{\sigma(i)} > v_{\sigma(j)}$ by definition of maximality, which implies $\sigma(j) > \sigma(i)$, showing that i is a sequential minimum of σ.

($\Longleftarrow$) For $(u_i, v_{\sigma(i)}) \in S$ such that i is a sequential minimum of σ. For any other point $(u_j, v_{\sigma(j)}) \in S$, $i \neq j$, either $i < j \Rightarrow u_i > u_j$ showing that $(u_i, v_{\sigma(i)})$ is maximal, or $j > i \Rightarrow \sigma(j) > \sigma(i)$ by definition of a sequential minimum, which implies $v_{\sigma(i)} > v_{\sigma(j)}$ showing that $(u_i, v_{\sigma(i)})$ is maximal. □

Corollary A1. *Membership in the Pareto set is invariant under strictly monotonic transformations of U or V.*

Proof. Strictly monotonic transformations leave the rank statistics unchanged and therefore also do not affect membership in the Pareto set by Lemma A1. □

We now turn to the main result of this Appendix: the proof of Theorem A1, which is restated here for convenience.

Theorem A1. *Let $S = \{(U_i, V_i)\}_{i=1}^N$ be a set of bivariate random variables drawn i.i.d. from a distribution with Lipschitz continuous CDF $F(u, v)$, and invertible marginal CDFs F_U, F_V. Define the region*

$$R_N \equiv \left\{ (u, v) \in [0,1] \times [0,1] : u + v - C(u, v) \geq e^{-\frac{1}{N}} \right\} \tag{A1}$$

where $C(u, v)$ denotes the copula of (U_i, V_i), which is the function that satisfies $F(u, v) = C(F_U(u), F_V(v))$.

Then, if the Lebesgue measure of this region $\lambda(R_N) = \Theta\left(\frac{\ell(N)}{N}\right)$ as $N \to \infty$, we have $\mathbb{E}\left[|\operatorname{Pareto}(S)|\right] = \Theta(\ell(N))$.

Proof. Since the marginal CDFs are invertible by assumption and therefore strictly monotonic, Corollary A1 allows us to consider instead $U'_i = F_U(U_i)$ and $V'_i = F_V(V_i)$ with the promise that $\operatorname{Pareto}(S') = \operatorname{Pareto}(S)$ where $S' \equiv \{(U'_i, V'_i\}$. Note that $F_{U'}(u') = u'$ and $F_{V'}(v') = v'$, and therefore without loss of generality, we can assume F_U and F_V are uniform distributions over the interval $[0, 1]$ dropping the prime notation. This allows us to identify the copula with the joint CDF $C(F_U(u), F_V(v)) = C(u, v) = F(u, v)$.

Let $\mathbf{1}_A(x)$ denote the indicator function of a set A: taking the value 1 for $x \in A$ and 0 otherwise. Then, $\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \mathbb{E}_S\left[\sum_{i=1}^N \mathbf{1}_{\operatorname{Pareto}(S)}(U_i, V_i)\right]$. Making use of the linearity of expectation and noting that (U_i, V_i) are drawn i.i.d., we can write

$$\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = N\mathbb{E}_S\left[\mathbf{1}_{\operatorname{Pareto}(S)}(U_1, V_1)\right] \tag{A2}$$

Note that $\mathbb{E}\left[\mathbf{1}_{\operatorname{Pareto}(S)}(u, v)\right] = (1 - \Pr[U > u, V > v])^N = (u + v - C(u, v))^N$, which follows from the definition of Pareto optimality. For convenience, we define $\hat{C}(u, v) \equiv u + v - C(u, v)$ yielding

$$\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \int_0^1 \int_0^1 Nf(u, v)\hat{C}(u, v)^{N-1} dudv \tag{A3}$$

Take $f_{\max}$ to be the maximum value f achieves over the domain, we are guaranteed $f_{\max} < \infty$ as C is Lipschitz by assumption. Therefore

$$\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] \leq Nf_{max} \int_0^1 \int_0^1 \hat{C}(u, v)^{N-1} dudv \tag{A4}$$

Now, define $\hat{C}_N$, which is equal to $\hat{C}$ in the region R_N and 0 otherwise. We also define the region

$$R'_N \equiv \left\{(u, v) \in [0, 1] \times [0, 1] : e^{-\frac{1+2\log N}{N}} \leq \hat{C}(u, v) < e^{-\frac{1}{N}}\right\} \tag{A5}$$

We now split the integral over $[0, 1]^2$ into three disjoint parts

$$\int_0^1 \int_0^1 \hat{C}(u, v)^{N-1} dudv = \int_{R_N} \hat{C}_N(u, v)^{N-1} dudv + \int_{R'_N} \hat{C}(u, v)^{N-1} dudv + \int_{[0,1]^2 \setminus R_N \cup R'_N} \hat{C}(u, v)^{N-1} dudv \tag{A6}$$

The integrand of the final term is bounded by $e^{-\log(N)+O(1)} = O(N^{-1})$ and $\lambda([0, 1]^2 \setminus R_N \cup R'_N) = \Theta(1)$; therefore, this term goes to 0 as $N \to \infty$. Now, we turn to the middle term on the right-hand side. Since C is 2-non-decreasing and Lipschitz, we have that the measure of the set $\lambda(R'_N) = \Theta\left(e^{-\frac{1}{N}} - e^{-\frac{1+2\log N}{N}}\right) = \Theta\left(\frac{\log N}{N}\right)$, $\hat{C}(u, v) < e^{-\frac{1}{N}}$ in the region R'_N by definition, and therefore, the second term goes to 0 as $N \to \infty$. Since there is always at least one point on the Pareto frontier, the first term must be $\Omega(1)$, and the integral is dominated by the portion over R_N. Equivalently,

$$\int_0^1 \int_0^1 \hat{C}(u, v)^{N-1} dudv \sim \int_0^1 \int_0^1 \hat{C}_N(u, v)^{N-1} dudv \tag{A7}$$

Further,

$$\int_0^1 \int_0^1 NC_N(u, v)^{N-1} dudv \leq N \int_0^1 \int_0^1 \mathbf{1}_{R_N}(u, v) dudv = N\lambda(R_N) = \ell(N) \tag{A8}$$

Following the chain of inequalities and asymptotic equivalences, we arrive at the desired result $\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \Theta(\ell(N))$. □

We now apply Theorem A1 to a few illustrative examples.

The Fréchet–Hoeffding copulae, W and M, are extremal in the sense that, written in *two* dimensions, any copula C must satisfy $W(u,v) \le C(u,v) \le M(u,v), \forall(u,v) \in [0,1]^2$; where $W(u,v) = \max(u+v-1,0)$ and $M(u,v) = \min(u,v)$. W and M correspond to complete negative and positive monotonic dependence, respectively.

Example A1 (Fréchet–Hoeffding lower bound)**.** *First, let us consider the scaling of the Pareto of a distribution with extremal copula $W(u,v)$. In this case, we note that the region $[0,1]^2 \setminus R_N$ is the triangle with vertices at $\{(0,0),(0,e^{-1/N}),(e^{-1/N},0)\}$, and therefore $\lambda(R_N) = 1 - \frac{1}{2}\exp^{-\frac{2}{N}}$. For large N, $\lambda(R_N) = \frac{1}{2} + O(N^{-1})$. We see that this satisfies the conditions for Theorem A1 with $\ell(N) = N$, giving $\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \Theta(N)$ as expected for a distribution with complete negative monotonic dependence.*

Example A2 (Fréchet–Hoeffding upper bound)**.** *First, let us consider the scaling of the Pareto of a distribution with extremal copula $M(u,v)$. In this case, we note that the region $[0,1]^2 \setminus R_N$ is the region $[0,e^{-1/N}]$, and therefore, $\lambda(R_N) = 1 - \exp^{-\frac{2}{N}}$. For large N, $\lambda(R_N) = \frac{2}{N} + O(N^{-2})$. We see that this satisfies the conditions for Theorem A1 with $\ell(N) = 1$, giving $\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \Theta(1)$ as expected for a distribution with complete positive monotonic dependence.*

Example A3 (Independent random variables)**.** *Next, let us consider the case of independent random variables with copula $C(u,v) = uv$. Note that the level curves in this case $u + v - C(u,v) = e^{-\frac{1}{N}}$ are given by $v = \frac{e^{-\frac{1}{N}}-u}{1-u}$. We can then integrate to find the area of the region R_N*

$$\lambda(R_N) = 1 - \int_0^{e^{-\frac{1}{N}}} \frac{e^{-\frac{1}{N}} - u}{1-u} du = e^{-1/n}\left(1 - e^{1/n}\right)\left(\log\left(1 - e^{-1/n}\right) - 1\right) \tag{A9}$$

Expanding for large N, we find that $\lambda(R_N) = \frac{\log N}{N} + O(N^{-1})$. We see that this satisfies the conditions for Theorem A1 with $\ell(N) = \log N$, giving $\mathbb{E}_S\left[|\operatorname{Pareto}(S)|\right] = \Theta(\log N)$.

Theorem A1 provides a useful tool to pin down the scaling of the size of the Pareto set. Due to the relatively quick decay of the additional terms in Equation (A6), we find that scaling estimates using the region R_N are quite accurate even for modest N. However, its applicability is limited, as it requires that we either have an analytic expression for the copula or are otherwise able to estimate the copula to precision $1/N$. In particular, we are not able to prove any bounds for the DIB frontier, which is the case $U = H(Z)$, and $V = I(Z;Y)$. We suspect that for most realistic datasets, including points on the DIB plane, that $\ell(N) = \operatorname{polylog}(N)$, which implies that the scaling of the Pareto set is likewise $\Theta(\operatorname{polylog}(N))$. Since we are interested in the large N behavior, we are hopeful that more general results can be found through the study of extreme-value copulas, which we leave for future work.

Appendix B. Auxiliary Functions

In this appendix, we provide the pseudocode for the important auxiliary functions used in Algorithms 1 and 2. The Pareto Set data structure is a list of point structures. A Point structure, p, contains fields for both objectives p.x, p.y, and optional fields for storing the uncertainties p.dx, p.dy and clustering function p.f. As a list, the Pareto Set P for Point p, and index i, also supports the functions $\textsc{size}(P)$ returning the number of elements in P, $\textsc{insert}(p,i,P)$ for inserting Point p at index i, and $\textsc{remove}(i,P)$ for removing the entry at index i. Additionally, since the Pareto Set P is maintained in sorted order by its first index, we can find the correct index at which to insert a new point in logarithmic time:

for a Point p and Pareto Set P, this is written FIND_INDEX$(p.x, P)$ in the pseudocode of Algorithms A1–A3.

Algorithm A1 Check if a point is Pareto optimal

Input: Point on objective plane p, and Pareto Set P
Output: TRUE if and only if p is Pareto optimal in P

```
procedure IS_PARETO($p, P$)
    $i$ = FIND_INDEX($p$.x, $P$)                 ▷ Return correct value to insert $p$ in $P$
    return SIZE($P$) = 0 or $i$ = SIZE($P$) or $P$[i + 1].y < $p$.y
```

Algorithm A2 Add point to Pareto Set

Input: Point on objective plane p, and Pareto Set P
Output: Updated Pareto Set P

```
procedure PARETO_ADD($p, P$)
    if IS_PARETO($p, P$) then                     ▷ Insert only if Pareto optimal
        $i$ = FIND_INDEX($p.x, P$)
        $P \leftarrow$ INSERT($p, i, P$)            ▷ Insert Point into correct location
        while $i <$ SIZE($P$) and $p$.y > $P$[i + 1].y do   ▷ Remove dominated points
            REMOVE($i + 1, P$)
            $i = i + 1$
    return P
```

Algorithm A3 Calculate distance to Pareto frontier

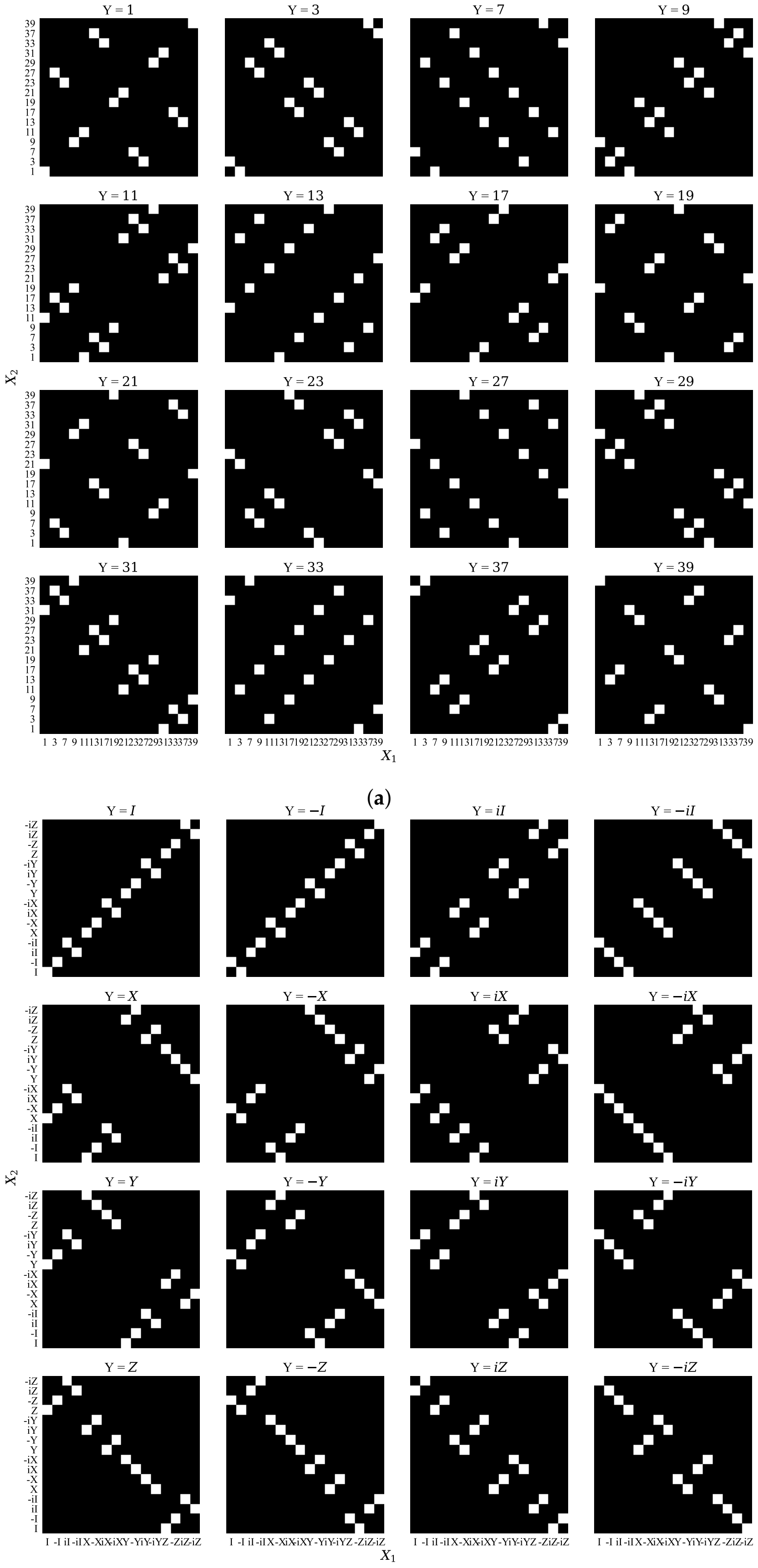
Input: Point on objective plane p, and Pareto Set P
Output: Distance to Pareto frontier (defined to be zero if Pareto optimal)

```
procedure PARETO_DISTANCE($p, P$)
    if IS_PARETO($p, P$) then return 0   ▷ Distance defined to be zero if Point is Pareto optimal
    $i$ = FIND_INDEX($p.x, P$)
    $d = P$[i].y − $p$.y                              ▷ Check top boundary
    while $P$[i].x − $p$.x < $d$ do
        if $i + 1 <$ SIZE($P$) and $P$[i].y > $p$.y then
            $q$ = POINT($x = P$[i].x, $y = P$[i + 1].y)
            $d$ = MINIMUM(DISTANCE($p, q$), $d$)           ▷ Check corners
        else
            $d$ = MINIMUM($P$[i].x − $p$.x, $d$)           ▷ Check right boundary
    return $d$
```

Appendix C. The Symmetric Pareto Mapper

In this appendix, we consider one way that Algorithm 1 can be modified to accommodate an additional structure in the dataset. The full pseudocode is provided in Algorithm A4 with the key difference occurring on line 12. This modification amounts to a redefining of the compressed variable $Z = (f(X_1), f(X_2))$. We would like to discover an encoding f that trades off the entropy of the encoding with the ability to predict Y from $(f(X_1), f(X_2))$. This corresponds to the following graphical model:

$$\begin{tikzcd}
Z_1 & X_1 \arrow[l, "f"'] \arrow[r] & Y \\
Z_2 & X_2 \arrow[l, "f"] \arrow[ur] &
\end{tikzcd}$$



(**a**)

(**b**)

Figure A1. Joint distribution $p_{X_1,X_2;Y}$ for the (**a**) $\left(\mathbb{Z}/40\mathbb{Z}\right)^{\times}$ group and (**b**) the Pauli group.

Algorithm A4 Symmetric Pareto Mapper

Input: Joint distribution $A, B, C \sim p_{ABC}$, and search parameter ε

Output: Approximate Pareto frontier P

```
procedure SYMMETRIC_PARETO_MAPPER($p_{ABC}, \varepsilon$)
    Pareto Set $P = \emptyset$                                   ▷ Initialize maintained Pareto Set
    Queue $Q = \emptyset$                                        ▷ Initialize search queue
    Point $p = (\mathrm{x} = -\mathrm{H}(p_{X_1X_2})/2, \mathrm{y} = \mathrm{I}(p_{X_1X_2;Y}), \mathrm{f} = \mathrm{id})$   ▷ Evaluate trivial clustering
    $P \leftarrow$ INSERT($p, P$)
    $Q \leftarrow$ ENQUEUE(id, $Q$)  ▷ Start with identity clustering id : $[n] \to [n]$ where $n = |X|$
    while $Q$ is not $\emptyset$ do
        $f =$ DEQUEUE($Q$)
        $n = |\mathrm{range}(f)|$
        for $0 < i < j < n$ do                                   ▷ Loop over all pairs of output clusters of $f$
            $f' = c_{i,j} \circ f$                               ▷ Merge clusters $i, j$ output $f$
            Point $p = \mathrm{Point}(\mathrm{x} = -\mathrm{H}(p_{f'(X_1)f'(X_2)})/2, \mathrm{y} = \mathrm{I}(p_{f'(X_1)f'(X_2);Y}), \mathrm{f} = f')$
            $d =$ PARETO_DISTANCE($p, P$)
            $P \leftarrow$ PARETO_ADD($p, P$)   ▷ Keep track of Point and clustering in Pareto Set
            with probability $e^{-d/\varepsilon}$, $Q \leftarrow$ ENQUEUE($f', Q$)

    return $P$
```

References

1. Cover, T.; Thomas, J. *Elements of Information Theory*; Wiley: Hoboken, NJ, USA, 2006.
2. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. *arXiv* **2000**, arXiv:physics/0004057.
3. Strouse, D.; Schwab, D.J. The deterministic information bottleneck. *Neural Comput.* **2017**, *29*, 1611–1630. [CrossRef] [PubMed]
4. Alemi, A.A.; Fischer, I. TherML: Thermodynamics of machine learning. *arXiv* **2018**, arXiv:1807.04162.
5. Fischer, I. The conditional entropy bottleneck. *Entropy* **2020**, *22*, 999. [CrossRef] [PubMed]
6. Hassanpour, S.; Wuebben, D.; Dekorsy, A. Overview and investigation of algorithms for the information bottleneck method. In Proceedings of the SCC 2017, 11th International ITG Conference on Systems, Communications and Coding, Hamburg, Germany, 6–9 February 2017; pp. 1–6.
7. Pereira, F.; Tishby, N.; Lee, L. Distributional clustering of English words. In Proceedings of the 31st Annual Meeting on Association for Computational Linguistics, Columbus, OH, USA, 22–26 June 1993; pp. 183–190.
8. Slonim, N.; Tishby, N. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems 12, Proceedings of the NIPS Conference, Denver, CO, USA, 29 November–4 December 1999*; Solla, S.A., Leen, T.K., Müller, K., Eds.; The MIT Press: Cambridge, MA, USA, 1999; pp. 617–623.
9. Banerjee, A.; Merugu, S.; Dhillon, I.S.; Ghosh, J.; Lafferty, J. Clustering with Bregman divergences. *J. Mach. Learn. Res.* **2005**, *6*, 1705–1749.
10. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep variational information bottleneck. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017 (Conference Track Proceedings, OpenReview.net), Toulon, France, 24–26 April 2017.
11. Andritsos, P.; Tsaparas, P.; Miller, R.J.; Sevcik, K.C. LIMBO: Scalable clustering of categorical data. In Proceedings of the Advances in Database Technology—EDBT 2004, 9th International Conference on Extending Database Technology, Crete, Greece, 14–18 March 2004; pp. 123–146. [CrossRef]
12. Kolchinsky, A.; Tracey, B.D.; Van Kuyk, S. Caveats for information bottleneck in deterministic scenarios. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
13. Tegmark, M.; Wu, T. Pareto-optimal data compression for binary classification tasks. *Entropy* **2019**, *22*, 7. [CrossRef] [PubMed]
14. Kurkoski, B.M.; Yagi, H. Quantization of binary-input discrete memoryless channels. *IEEE Trans. Inf. Theory* **2014**, *60*, 4544–4552. [CrossRef]
15. Zhang, J.A.; Kurkoski, B.M. Low-complexity quantization of discrete memoryless channels. In Proceedings of the 2016 International Symposium on Information Theory and Its Applications (ISITA), Monterey, CA, USA, 30 October–2 November 2016; pp. 448–452.
16. Navon, A.; Shamsian, A.; Fetaya, E.; Chechik, G. Learning the Pareto Front with Hypernetworks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 36–30 April 2020.
17. Lin, X.; Yang, Z.; Zhang, Q.; Kwong, S. Controllable pareto multi-task learning. *arXiv* **2020**, arXiv:2010.06313.
18. Strouse, D.; Schwab, D.J. The information bottleneck and geometric clustering. *Neural Comput.* **2019**, *31*, 596–612. [CrossRef] [PubMed]
19. Still, S.; Bialek, W. How many clusters? An information-theoretic perspective. *Neural Comput.* **2004**, *16*, 2483–2506. [CrossRef] [PubMed]

20. Awasthi, P.; Charikar, M.; Krishnaswamy, R.; Sinop, A.K. The hardness of approximation of euclidean k-Means. In Proceedings of the 31st International Symposium on Computational Geometry (SoCG 2015), Eindhoven, The Netherlands, 22–25 June 2015.
21. Nemenman, I.; Shafee, F.; Bialek, W. Entropy and Inference, Revisited. In *Advances in Neural Information Processing Systems*; Dietterich, T., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2002; Volume 14, pp. 471–478.
22. Paninski, L. Estimation of entropy and mutual information. *Neural Comput.* **2003**, *15*, 1191–1253. [CrossRef]
23. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138. [CrossRef] [PubMed]
24. Nemenman, I.; Bialek, W.; de Ruyter van Steveninck, R. Entropy and information in neural spike trains: Progress on the sampling problem. *Phys. Rev. E* **2004**, *69*, 056111. [CrossRef] [PubMed]
25. Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; Tucker, G. On variational bounds of mutual information. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 5171–5180.
26. Twomey, C.R.; Roberts, G.; Brainard, D.H.; Plotkin, J.B. What we talk about when we talk about colors. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2109237118. [CrossRef] [PubMed]
27. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2014; pp. 740–755.
28. Achille, A.; Soatto, S. Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res.* **2018**, *19*, 1947–1980.
29. Wu, T.; Fischer, I.S. Phase Transitions for the Information Bottleneck in Representation Learning. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020 (OpenReview.net), Addis Ababa, Ethiopia, 26–30 April 2020.
30. Wu, T.; Fischer, I.; Chuang, I.L.; Tegmark, M. Learnability for the information bottleneck. In *Uncertainty in Artificial Intelligence, Proceedings of the PMLR, Cambridge, MA, USA, 16–18 November 2020*; AUAI Press: Corvallis, OR, USA, 2020; pp. 1050–1060.
31. Ngampruetikorn, V.; Schwab, D.J. Perturbation theory for the information bottleneck. In *Advances in Neural Information Processing Systems*; Springer: Berlin, Germany, 2021; Volume 34.
32. Rezende, D.J.; Viola, F. Taming VAEs. *arXiv* **2018**, arXiv:1810.00597.
33. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jeju Island, Korea, 11–15 October 2015; pp. 1–5.
34. Chaudhari, P.; Soatto, S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In Proceedings of the 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 11–16 February 2018; pp. 1–10.
35. Saxe, A.M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B.D.; Cox, D.D. On the information bottleneck theory of deep learning. *J. Stat. Mech. Theory Exp.* **2019**, *2019*, 124020. [CrossRef]
36. Still, S. Thermodynamic cost and benefit of memory. *Phys. Rev. Lett.* **2020**, *124*, 050601. [CrossRef] [PubMed]
37. Buesing, L.; Maass, W. A spiking neuron as information bottleneck. *Neural Comput.* **2010**, *22*, 1961–1992. [CrossRef] [PubMed]