Error Slice Discovery via Manifold Compactness

Han Yu[1], Hao Zou[1], Jiashuo Liu[1], Renzhe Xu[2], Yue He[3], Xingxuan Zhang[1], Peng Cui[1]*

[1]Tsinghua University
[2]Shanghai University of Finance and Economics
[3]Renmin University of China
yuh21@mails.tsinghua.edu.cn, ahio@163.com, liujiashuo77@gmail.com, xurenzhe@sufe.edu.cn
hy865865@gmail.com, xingxuanzhang@hotmail.com, cuip@tsinghua.edu.cn

Abstract

Despite the great performance of deep learning models in many areas, they still make mistakes and underperform on certain subsets of data, i.e. *error slices*. Given a trained model, it is important to identify its semantically coherent error slices that are easy to interpret, which is referred to as the *error slice discovery* problem. However, there is no proper metric of slice *coherence* without relying on extra information like predefined slice labels. Current evaluation of slice coherence requires access to predefined slices formulated by metadata like attributes or subclasses. Its validity heavily relies on the quality and abundance of metadata, where some possible patterns could be ignored. Besides, current algorithms cannot directly incorporate the constraint of coherence into their optimization objective due to absence of an explicit coherence metric, which could potentially hinder their effectiveness. In this paper, we propose *manifold compactness*, a coherence metric without reliance on extra information by incorporating the data geometry property into its design, and experiments on typical datasets empirically validate the rationality of the metric. Then we develop Manifold Compactness based error Slice Discovery (MCSD), a novel algorithm that directly treats risk and coherence as the optimization objective, and is flexible to be applied to models of various tasks. Extensive experiments on the benchmark and case studies on other typical datasets demonstrate the superiority of MCSD.

Introduction

In recent years, with the enhancement of computational power, neural networks have achieved significant progress in numerous tasks (Achiam et al. 2023; Liu et al. 2023a; Tian, Ye, and Doermann 2025). Despite their impressive overall performance, they are far from perfect, and still suffer from performance degradation on some subpopulations (Yang et al. 2023). This substantially hinders their application in risk-sensitive scenarios like medical imaging (Yang et al. 2024), autonomous driving (Chen et al. 2024), etc., where model mistakes may result in catastrophic consequences. Therefore, to avoid the misuse of models, it is a fundamental problem to identify subsets (or slices) where a given model tends to underperform. Moreover, we would like to find coherent interpretable semantic patterns in the underperforming slices. For example, a facial recognition model may underperform in certain demographic groups like elderly females. An autonomous driving system may fail in the face of steep road conditions. Identifying such coherent patterns could help us understand model failures, and we could employ straightforward solutions for improvement like collecting new data (Liu et al. 2023b) or upweighting samples in error slices (Liu et al. 2021).

Previously, works of *error slice discovery* (d'Eon et al. 2022; Wang et al. 2023) aim for this goal. Despite the emphasis on coherence in error slice discovery, there is no proper metric to assess the coherence of a given slice without additional information like predefined slice labels. On one hand, this impairs the efficacy of the evaluation paradigm of error slice discovery. In the current benchmark (Eyuboglu et al. 2022), with the help of metadata like attributes or subclasses, it predefines slices that are already semantically coherent, and they depict the coherence of a slice discovered by a specific algorithm via the matching degrees between it and the predefined underperforming slices, so as to evaluate the effectiveness of the algorithm. Such practice heavily relies on not only the availability but also the quality of metadata, whose annotations are usually expensive, and may overlook model failure patterns not captured by existing metadata. On the other hand, due to the absence of an explicit coherence metric, current algorithms can only indirectly incorporate the constraint of coherence into their design, e.g. via clustering (Eyuboglu et al. 2022; Wang et al. 2023; Plumb et al. 2023), without treating it as a direct optimization objective. This could potentially impede the development of more effective error slice discovery algorithms.

In this paper, inspired by the data geometry property that high dimensional data tends to lie on a low-dimensional manifold (Belkin and Niyogi 2003; Roweis and Saul 2000; Tenenbaum, Silva, and Langford 2000), we incorporate this property to propose *manifold compactness* as the metric of coherence given a slice, which does not require additional information. We illustrate the validity of the metric by showing that it captures semantic patterns better than depicting coherence via metrics directly calculated in Euclidean space, and is empirically consistent with current evaluation metrics that require predefined slice labels. Then we propose a novel and flexible algorithm named Manifold Compactness based error Slice Discovery (MCSD) that jointly optimizes

*Corresponding author

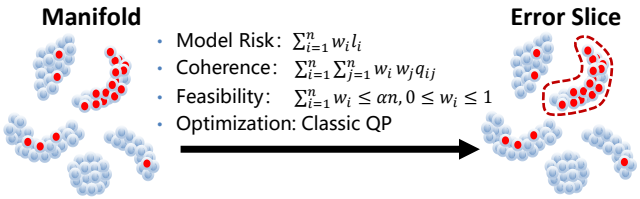


Figure 1: Illustration of MCSD. Blue points are correctly classified by the given trained model, while red ones are wrongly classified. The trained model achieves a good overall accuracy, but exhibit a high error in a certain slice.

the average risk and manifold compactness to identify the error slice. Thus both the risk and coherence, i.e. the desired properties of error slices are explicitly treated as the optimization objective. We illustrate our algorithm in Figure 1. Besides, our algorithm can be directly applied to trained models of different tasks while most error slice discovery methods are restricted to classification only. We provide theoretical analyses of the algorithm. We conduct experiments on dcbench (Eyuboglu et al. 2022) to demonstrate our algorithm's superiority compared with existing ones. We also provide several case studies on different types of datasets and tasks to showcase the effectiveness and flexibility of our algorithm. Our contributions are summarized below:

- We define manifold compactness as the metric of slice coherence without additional information. We empirically show that it captures semantic patterns well, proving its rationality.
- We propose MCSD, a flexible algorithm that directly incorporates the desired properties of error slices, i.e. risk and coherence, into the optimization objective. It can also be applied to trained models of various tasks.
- We provide theoretical analyses of the algorithm. We conduct experiments on the error slice discovery benchmark to show that our algorithm outperforms existing ones, and we perform diverse case studies to demonstrate the usefulness and flexibility of our algorithm.

Problem

Due to space limit, we leave the section of related works in Appendix. Unless stated otherwise, for random variables, we use uppercase letters, in contrast to a concrete dataset where we use lowercase letters. Consider classic supervised learning. The input variable is denoted as $X \in \mathcal{X}$ and the outcome is denoted as $Y \in \mathcal{Y}$, whose joint distribution is $P(X, Y)$. There exist multiple slices, where j-th slice can be represented as a slice label variable $S^{(j)} \in \{0, 1\}$. For classic supervised learning, the goal is to learn a model $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$ with parameter θ. Denote $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto [0, +\infty]$ as the loss function. Current machine learning algorithms are capable of learning models with a satisfying overall performance, which can be demonstrated via a low risk $\mathbb{E}_P[\ell(f_\theta(X), Y)]$ over the whole population. However, performance degradation could still occur in a certain subpopulation or slice. Here we introduce error slice discovery:

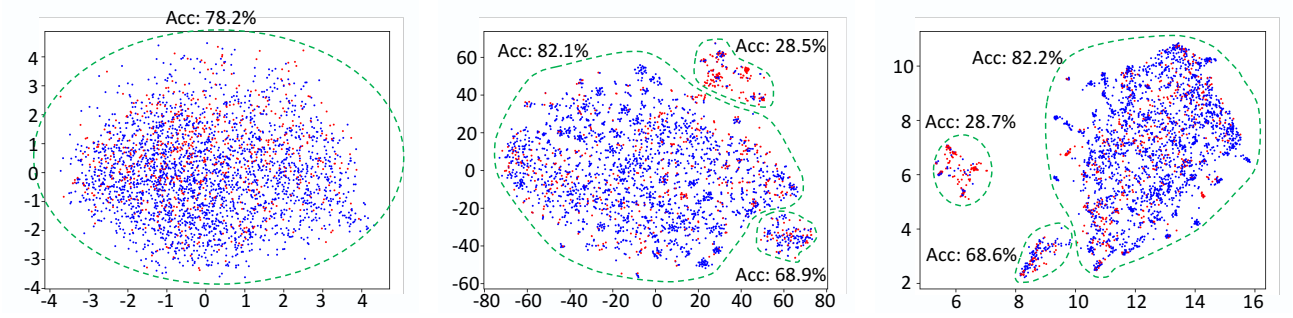
Problem 0.1 (Error Slice Discovery)**.** Given a fixed prediction model $f_{\theta_0} : \mathcal{X} \mapsto \mathcal{Y}$ and a validation dataset $\mathcal{D}_{\text{va}} = \{(x_i^{\text{va}}, y_i^{\text{va}})\}_{i=1}^{n_{\text{va}}}$, we aim to develop an algorithm $\mathcal{A}$ that takes $\mathcal{D}_{\text{va}}$ and f_{θ_0} as input, and learns slicing functions $g_\varphi^{(j)} : \mathcal{X} \times \mathcal{Y} \mapsto \{0, 1\}, 1 \le j \le K$. Denote the output of j-th slicing function as $\hat{S}_j$. We require that the risk in the slice is higher than the population-level risk by a certain threshold: $\mathbb{E}_{X,Y\sim P(X,Y|\hat{S}_j=1)}[\ell(f_\theta(X), Y)] > \mathbb{E}_{X,Y\sim P(X,Y)}[\ell(f_\theta(X), Y)] + \epsilon$, and the discovered slice is as coherent as possible for convenience of interpretation.

The reason why we require an extra validation dataset to implement error slice discovery is that for deep learning models, training data is usually fitted well enough or even nearly perfect. Thus model mistakes on training data carry much less information on models' generalization ability. This is common practice in previous works (d'Eon et al. 2022; Eyuboglu et al. 2022; Wang et al. 2023). Without ambiguity, we omit the superscript or subscript of "va" for n, x_i, y_i for convenience in the next two sections.

Metric

Due to the absence of a proper metric for coherence that is independent of additional information, the current benchmark (Eyuboglu et al. 2022) provides numerous datasets, trained models, and their predefined underperforming slice labels. They employ precision@k, i.e. the proportion of the top k elements in the discovered slice belonging to the predefined ground-truth error slice as the metric of slice coherence to evaluate error slice discovery algorithms. Although such practice is reasonable to some extent, its effectiveness of evaluation strongly relies on the quality of metadata that composes the underperforming slice labels, which might be not even available under many circumstances.

To eliminate the requirement of predefined slices, we try to propose a new metric of coherence. It is commonly acknowledged that high-dimensional data usually lies on a low-dimensional manifold (Belkin and Niyogi 2003; Roweis and Saul 2000; Tenenbaum, Silva, and Langford 2000). In this case, while direct usage of Euclidean distance cannot properly capture the dissimilarity between data points, the geodesic distance in the metric space of the manifold can. For preliminary justification, here we provide visualization analyses based on different types of dimension-reduction techniques. Among these techniques, PCA mainly preserves pairwise Euclidean distances between data points while t-SNE and UMAP are both manifold learning techniques. In Figure 2, blue dots are correctly classified by the trained model and red dots are wrongly classified. We can see that the visualization of t-SNE and UMAP shows much clearer clustering structures than that of PCA, either having a larger number of clusters or exhibiting larger margins between clusters. This indicates that it could be better to measure coherence in the metric space of a manifold than in the original Euclidean space. Due to space limit, we only present results of the widely adopted facial dataset CelebA (Liu et al. 2015) here, leaving results of other datasets in Appendix, where the same conclusion holds.



(a) PCA (b) t-SNE (c) UMAP

Figure 2: Category "Blond Hair" of CelebA. Visualization of t-SNE and UMAP (manifold-based dimension reduction) shows much clearer clustering structures than that of PCA (mainly preserving Euclidean distances between data points). Thus it could be better to measure coherence in the metric space of a manifold than using metrics directly calculated in Euclidean space.

Therefore, we attempt to define a metric of coherence inside the discovered slice via compactness in the data manifold. In practice, a manifold can be treated as a graph G (Melas-Kyriazi 2020), and we can apply graph learning methods like k-nearest neighbor (kNN) to approximate it (Dann et al. 2022). Given an identified slice $\hat{\mathcal{S}} = \{(x_i, y_i)|\hat{s}_i = 1\}$, where $\hat{s}_i$ is the output of the slicing function on i-th sample, we define manifold compactness as:

Definition 0.2 (Manifold Compactness)**.** Consider a given approximation of the data manifold, i.e. a weighted graph $G = (V, E, Q)$. The node set $V = \{v_i\}_{i=1}^n$ corresponds to the dataset $\{(x_i, y_i)\}_{i=1}^n$. The edge set $E = \{e_{ij}\}_{1\le i,j\le n}$, where e_{ij} represents whether node v_i and v_j are connected in the graph G. The weights $Q = \{q_{ij}\}_{1\le i,j\le n}$, where q_{ij} represents the weight of edge e_{ij}. Given a slice $\hat{\mathcal{S}}$, the manifold compactness of it can be defined as:

$$\mathrm{MC}(\hat{\mathcal{S}}) = \frac{1}{|\hat{\mathcal{S}}|} \sum_{(x_i,y_i),(x_j,y_j)\in\hat{\mathcal{S}}} q_{ij} \tag{1}$$

This metric is the average weighted degree of nodes of the induced subgraph, whose vertex set corresponds to the slice. The higher it is, the denser or more compact the subgraph is, implying a more coherent slice. Note that when applying this to evaluate multiple slice discovery algorithms, for convenience of comparison, we control the size of $\hat{\mathcal{S}}$ for those algorithms to be the same by taking the top αn data points sorted by the slicing function's prediction probability. Here n is the size of the dataset and $\alpha \in (0, 1]$ is a fixed proportion. The operation of selecting data points with highest prediction probabilities is akin to calculating precision@k in dcbench (Eyuboglu et al. 2022).

Next, we try to demonstrate the validity and advantages of our proposed coherence metric. A common and representative metric of coherence directly calculated in Euclidean space is variance. Thus we measure variance and manifold compactness respectively on different semantically predefined slices of CelebA (Liu et al. 2015). We use the binary label y to indicate whether the person has blond hair or not,

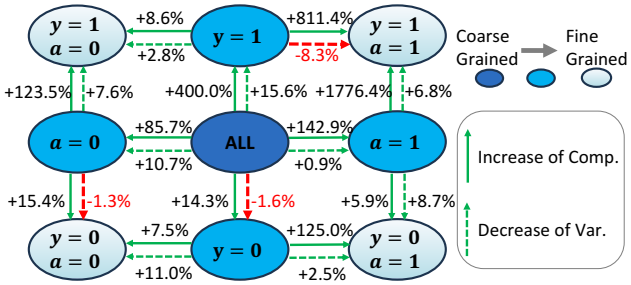


Figure 3: Percentage of increase of manifold compactness ("Comp.") and decrease of variance ("Var.") from coarse-grained slices to fine-grained ones in CelebA. For manifold compactness, there is always a positive increase from coarse-grained slices to fine-grained slices. However, in some cases, variance fails to decrease from coarse-grained slices to fine-grained slices as expected, which are marked in red arrows. This could imply that manifold compactness is better at capturing semantic coherence than variance does.

and a to indicate whether the person is male or not. The values of y and a can formulate slices of different granularity. In Figure 3, the most coarse-grained slice is the whole dataset (the darkest circle in the center), the most fine-grained slice is the combination of y and a (the lightest circles in the four corners), and slices of the middle granularity are formulated by either of y and a. Figure 3 shows the percentage of the increase of manifold compactness and the decrease of variance with directed arrows from semantically coarse-grained slices to fine-grained ones. It is intuitive that these digits are supposed to be positive if these two metrics could properly measure semantic coherence. However, for variance, in some cases the value of the more coarse-grained slice is even smaller than the more fine-grained, marked in red arrows. For manifold compactness, there is always a positive increase from semantically coarse-grained slices to fine-grained slices. In this way, we demonstrate that manifold compactness is better at capturing semantic coherence than variance does. Still due to the space limit, we only provide

Algorithm 1: Manifold Compactness based Error Slice Discovery (MCSD)

Input:
Validation dataset: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$.
The trained model to be evaluated: $f_{\theta_0} : \mathcal{X} \mapsto \mathcal{Y}$.
Size of the slice as a proportion of the dataset: α.
Coherence coefficient λ.
A pretrained feature extractor: $h_{fe} : \mathcal{X} \mapsto \mathcal{Z}$.
Output: The identified error slice $\hat{\mathcal{S}}$.
for $i = 1$ to n **do**
  Calculate the embedding: $z_i = h_{fe}(x_i)$.
  Calculate model prediction loss: $l_i = \ell(f_{\theta_0}(x_i), y_i)$.
end for
Establish the kNN graph $G = (V, E, Q)$ based on the embeddings $\{z_i\}_{i=1}^n$.
Formulate the quadratic programming problem with variables $\{w_i\}_{i=1}^n$ as Equation (2).
Employ Gurobi to solve the problem in Equation (2).
for $i = 1$ to n **do**
  $\hat{s}_i = 1$ **if** $w_i > \alpha$-Quantile of $\{w_i\}_{i=1}^n$ **else** 0.
end for
return: $\hat{\mathcal{S}} = \{(x_i, y_i) | \hat{s}_i = 1\}$

results of CelebA here, and leave detailed values and results of other datasets, along with comparisons against other metrics directly calculated in Euclidean space like Median Absolute Deviation (MedianAD) in Appendix, where we reach the same conclusion. Besides, in Table 1 of Section Experiments, we have also empirically shown that the rank order of the four methods according to precision metrics is generally the same as that of manifold compactness. Since the precision metrics are based on predefined slice labels with semantic meanings, it implies that our proposed coherence metric could capture semantic patterns well and is appropriate for evaluation of slice discovery algorithms even when predefined slice labels are absent.

Algorithm

We introduce Manifold Compactness based error Slice Discovery (MCSD), a novel error slice discovery algorithm that incorporates the data geometry property by taking manifold compactness into account. In this way, the metrics of both risk and coherence can be treated as the explicit objective of optimization, thus better enabling the identified error slice to exhibit consistent and easy-understanding semantic meanings. The detailed algorithm is described in Algorithm 1. It is worth noting that although we mainly focus on the identified worst-performing slice for convenience of analyses and comparison, our algorithm could discover more error slices by removing the first discovered slice from the validation dataset and applying our algorithm repeatedly to the rest of the dataset for more error slices. Related experiments and analyses are included in Appendix.

First, we approximate the data manifold via a graph. To facilitate the graph learning approach, we obtain the embeddings of the dataset via a pretrained feature extractor (Radford et al. 2021), i.e. $z_i = h_{fe}(x_i)$, which follows previous works of error slice discovery (Eyuboglu et al. 2022; Wang et al. 2023). Then we construct a kNN graph $G = (V, E, Q)$ based on the embeddings $\{z_i\}_{i=1}^n$, which is a widely adopted manifold learning approach (Zemel and Carreira-Perpiñán 2004; Pedronette, Gonçalves, and Guilherme 2018; Dann et al. 2022). In the graph G, the edge weight $q_{ij} = 1$ if z_j is among the k nearest neighbors of z_i, or else $q_{ij} = 0$.

For the convenience of optimization, instead of direct hard selection, we assign a sample weight w_i for each (x_i, y_i), which is the variable to be optimized and is restricted in the range $[0, 1]$. We theoretically prove the equivalence between hard selection and sample weight optimization in Appendix. Considering the model risk, we employ the weighted average mean of loss $\sum_{i=1}^n w_i l_i$ as our optimization objective, where $l_i = \ell(f_{\theta_0}(x_i), y_i)$ is the model prediction loss of ith sample given f_{θ_0}. Considering coherence, we adopt manifold compactness in Definition 0.2 as the optimization objective, i.e. $\sum_{i=1}^n \sum_{j=1}^n w_i w_j q_{ij}$. We add these two objectives with a hyperparameter λ. We also restrict the size of the identified slice to be no more than a proportion α of the dataset. Thus we formulate the optimization problem as a quadratic programming (QP) problem:

$$\begin{aligned} \max_{\{w_i\}_{i=1}^n} & \sum_{i=1}^n w_i l_i + \lambda \sum_{i=1}^n \sum_{j=1}^n w_i w_j q_{ij} \\ s.t. & \sum_{i=1}^n w_i \leq \alpha n \\ & 0 \leq w_i \leq 1, \quad \forall 1 \leq i \leq n \end{aligned} \tag{2}$$

The above QP problem can be solved by classic optimization algorithms or powerful mathematical optimization solvers like Gurobi (Gurobi Optimization 2021). After solving for the proper sample weights $\{w_i\}_{i=1}^n$, we select the top αn samples sorted by the weights as the error slice $\hat{\mathcal{S}}$. Note that in most previous algorithms' workflow, they require prediction probabilities as input (Eyuboglu et al. 2022; Plumb et al. 2023; Wang et al. 2023), thus only applicable to classification, while our algorithm takes prediction loss as input, naturally more flexible and applicable to various tasks.

Experiments

In this section, we conduct extensive experiments to demonstrate the validity of our proposed metric and the advantages of our algorithm MCSD compared with previous methods. For quantitative results, we conduct experiments on the error slice discovery benchmark *dcbench* (Eyuboglu et al. 2022). Besides, we conduct experiments on other types of datasets like classification for medical images (Irvin et al. 2019), object detection for driving (Yu et al. 2020), and detection of toxic comments (Borkan et al. 2019), which showcase the great potential of our algorithm to be applied to various tasks. The baselines we compare with are Spotlight (d'Eon et al. 2022), Domino (Eyuboglu et al. 2022), and PlaneSpot (Plumb et al. 2023). More experimental details are included in Appendix.

For evaluation, we compute manifold compactness as the main coherence metric along with the average performance

Metric	Precision@10 (%) ↑			Precision@25 (%) ↑			Average Precision (%) ↑			Manifold Comp.↑		
Method	Corr.	Rare	Noisy	Corr.	Rare	Noisy	Corr.	Rare	Noisy	Corr.	Rare	Noisy
Spotlight	32.3	28.7	43.2	32.2	26.4	40.9	28.9	16.4	22.7	4.78	2.67	4.20
Domino	36.2	52.5	51.7	33.8	52.3	50.0	29.9	37.7	31.3	4.14	4.06	5.53
PlaneSpot	26.1	18.1	29.4	22.3	18.1	27.8	21.8	14.3	18.8	2.93	1.59	3.30
MCSD	**47.4**	**61.1**	**60.6**	**45.6**	**59.8**	**57.4**	**40.3**	**52.4**	**38.4**	**6.22**	**7.81**	**8.71**

Table 1: Results of dcbench. We mark the best method in bold type and underline the second-best in terms of each metric. "Comp." means "Manifold Compactness". "Corr." means "Correlation". "%" indicates the digits are percentage values.

of the given model f_{θ_0} on the identified slice $\hat{S}$. For classification, the performance metric is accuracy. For object detection, it is Average Precision (AP). Note that there are two aspects of evaluation simultaneously. In this case, we put more emphasis on coherence than performance, since we only require the performance of the identified slice to be low to a certain degree but expect it to be as coherent as possible for the benefits of interpretation. This is similar to dcbench (Eyuboglu et al. 2022) where coherence outweighs performance and is chosen as the main evaluation metric.

For running time comparison and related analyses of our method and the baselines, we leave results in Appendix. For the choice and analyses of hyperparameters, we leave them in Appendix. For the improvement of the original models utilizing the discovered error slices, we leave results in Appendix. Due to space limit, we put more examples including those of Spotlight and PlaneSpot in Appendix, about 20 images for each identified slice.

Benchmark Results: Dcbench

Dcbench (Eyuboglu et al. 2022) offers 886 publicly available settings for error slice discovery. Each setting consists of a trained ResNet-18 (He et al. 2016), a validation dataset and a test dataset, both with predefined underperforming slice labels. The validation dataset and its error slice labels are taken as input of slice discovery methods, while the test dataset and its error slice labels are used for evaluation. There are three types of slices in dcbench: correlation slices, rare slices, and noisy label slices. The correlation slices are generated from CelebA (Liu et al. 2015), while the other two types of slices are generated from ImageNet (Deng et al. 2009). More details are included in Appendix. In terms of evaluation metrics, we employ precision@k and average precision following dcbench's practice, where precision@k is the proportion of samples with top k highest probabilities output by the learned slicing function that belongs to the predefined underperforming slice, and average precision is calculated based on precision@k with different values of k. We also calculate manifold compactness as Definition 0.2. For all these metrics, a higher value indicates higher coherence of the identified slice, thus implying a more effective algorithm capable of error slice discovery.

Effectiveness of our method Table 1 shows that MCSD outperforms other methods across all three types of error slices in precision@10, precision@25, average precision, and manifold compactness. This greatly exhibits the strengths of our method compared with existing ones in error slice discovery. Among the baselines, Domino consistently ranks 2nd, also showing a fair performance.

Validity of our metric It is also worth noting that the proposed metric manifold compactness shows a strong consistency with other metrics. Table 1 shows that the rank order of the four methods based on precision metrics is usually MCSD, Domino, Spotlight, PlaneSpot, the same as the rank order based on manifold compactness, except for the correlation slice where the rank order of Domino and Spotlight switches. While other metrics require access to predefined labels of underperforming slices, our metric does not. This demonstrates the validity and advantages of our proposed manifold compactness when measuring coherence and evaluating error slice discovery algorithms.

Case Study: CelebA

CelebA (Liu et al. 2015) is a large facial dataset of 202,599 images, each with annotations of 40 binary attributes. In the setting of subpopulation shift, it is the most widely adopted dataset since it is easy to generate spurious correlations between two specific attributes by downsampling the dataset (Yang et al. 2023; Sagawa et al. 2019; Liu et al. 2021). Different from settings in dcbench, in this case study we follow Sagawa et al. (2019) to treat the binary label of blond hair as the target of prediction and directly use the whole dataset of CelebA (Liu et al. 2015) without downsampling, thus closer to the real scenario. In terms of implementation details, we employ the default data split provided by CelebA and follow the training process of ERM in (Sagawa et al. 2019) to train a ResNet-50. We apply error slice discovery algorithms on both categories respectively, thus taking advantage of outcome labels that are known during slice discovery. We also illustrate results of directly selecting top αn_{te} samples sorted by prediction losses.

From Table 2, we can see that for both categories of CelebA, our algorithm identifies the most coherent underperforming slice in terms of manifold compactness, where higher is better. Although it ranks 2nd for the category of blond hair in terms of accuracy, where lower is better, for the task of error slice discovery, we put more emphasis on coherence since we want the identified slices to be interpretable, and we only require the performance of the slice to be lower than a threshold compared with the overall performance. In Figure 4, left five columns and right five columns are from two categories separately. Four rows correspond to

Blond Hair?	Yes		No	
Method	Acc. (%) ↓	Comp.↑	Acc. (%) ↓	Comp.↑
Spotlight	**26.3**	5.71	**65.9**	3.35
Domino	34.6	6.07	82.1	3.58
PlaneSpot	68.4	2.92	93.6	1.13
MCSD	33.8	**8.09**	75.7	**5.54**
Overall	76.4	-	98.2	-

Table 2: Results on CelebA and the overall accuracy of the trained model. "Acc." means "Accuracy". "Comp." means "Manifold Compactness". We mark the best method in bold type and underline the second-best. "%" indicates the digits are percentage values.

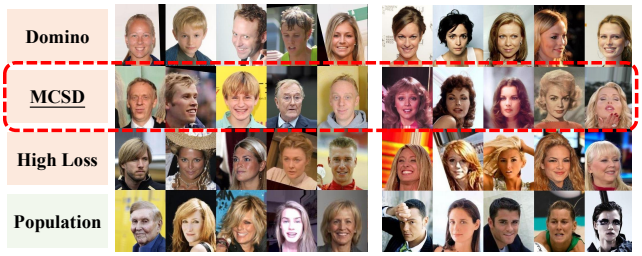


Figure 4: Images randomly sampled from slices of CelebA. Left five columns are results of the category "Blond Hair". Right five columns are results of the category "Not Blond Hair". We can see that MCSD is capable of finding error slices that are more coherent than others.

randomly sampled images from different sources: the error slice that Domino identifies, the error slice that MCSD identifies, top αn_{te} samples sorted by the loss, and all samples of the corresponding category. We can see that images from the error slice identified by MCSD obviously exhibit more coherent characteristics than others.

For the category of blond hair, images in the row of MCSD are all faces of males, conforming to the intuition that models may learn the spurious correlation between blond hair and female, and could be inclined to make mistakes in subgroups like males with blond hair in the row of MCSD. Although more than half of the images for Domino in the blond hair category are also males, its coherence is much smaller than that of MCSD, making it hard for humans to interpret the failure pattern when compared with images of the whole population. Besides, in the third row, when simply taking account of the prediction loss to select risky samples, it is also difficult to extract the common pattern. For the category of not blond hair, although both Domino and sorting-by-loss can extract the pattern of faces being female with brown hair or blond hair (label noise), MCSD identifies more detailed common characteristics that faces in the images are not only female, but bear vintage styles like in the 20th century, which also constitute a riskier slice than Domino in terms of accuracy. It is also worth noting that MCSD achieves a higher manifold compactness than Domino in Table 2, consistent with that the identified slice of MCSD exhibits more coherent semantics in Figure 4, further confirming the rationality of our coherence metric.

Ill?	Yes		No	
Method	Acc. (%) ↓	Comp.↑	Acc. (%) ↓	Comp.↑
Spotlight	**19.5**	2.10	64.9	4.70
Domino	31.5	1.53	88.4	2.82
PlaneSpot	42.8	3.66	69.5	3.17
MCSD	31.5	**4.70**	**63.3**	**4.87**
Overall	45.5	-	91.0	-

Table 3: Results on CheXpert and overall accuracy of the trained model. "Acc." means "Accuracy". "Comp." means "Manifold Compactness". We mark the best method in bold type and underline the second-best. "%" indicates the digits are percentage values.

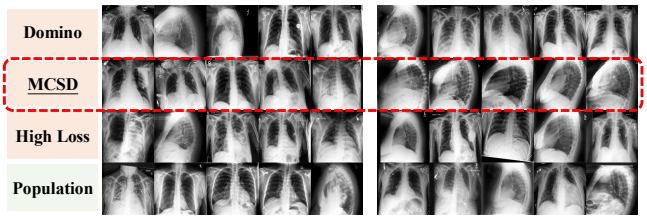


Figure 5: Images randomly sampled from slices of CheXpert. Left five columns are results of the category "Ill". Right five columns are results of the category "Healthy". We can see that MCSD is capable of finding error slices that are more coherent than others.

Case Study: CheXpert

To demonstrate the effectiveness of our algorithm on other types of data, we conduct experiments on a medical imaging dataset, i.e. CheXpert (Irvin et al. 2019), where the task is to predict whether patients are ill or not based on their chest X-ray images. It contains 224,316 images from 65,240 patients. We follow the data split and training process of Yang et al. (2023) to train a ResNet-50. Still, we apply algorithms to images of ill and healthy patients respectively.

In Table 3, we can see that MCSD still achieves highest manifold compactness and relatively low slice accuracy in terms of the discovered error slice for both ill and healthy patients. In Figure 5, for ill patients, images sampled from the error slice discovered by MCSD are all taken from the frontal view, while there are different views for images sampled from other sources. For healthy patients, images corresponding to MCSD are all taken from the left lateral view, while other rows constitute images from different views, making it difficult to extract the common risky pattern. These results showcase MCSD's usefulness in medical imaging, which is a highly risk-sensitive task and deserves more attention for error slice discovery and failure pattern interpretation. Besides, the consistency of the order of coherence for MCSD and Domino in Table 3 and Figure 5 also confirms the rationality of our proposed coherence metric.

Case Study: BDD100K

Category	Pedestrian		Traffic Light	
Method	AP (%) ↓	Comp.↑	AP (%) ↓	Comp.↑
Spotlight	57.3	2.05	**46.3**	2.61
MCSD	**53.8**	**6.60**	57.3	**4.78**
Overall	71.4	-	69.2	-

Table 4: Results of algorithms on BDD100K for two categories, along with the overall AP of the trained model. "Comp." means "Manifold Compactness". We mark the best method in bold type. "%" indicates that the digits are percentage values.

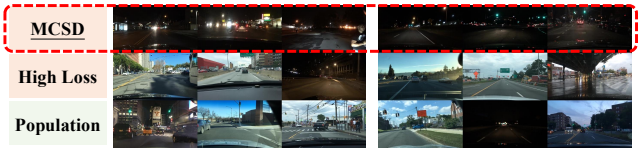


Figure 6: Images randomly sampled from slices of BDD100K. Left three columns are results of the category "Pedestrian". Right three columns are results of the category "Traffic Light". It shows that MCSD can find error slices that are more coherent than others.

Compared with most previous algorithms (Eyuboglu et al. 2022; Wang et al. 2023; Plumb et al. 2023) requiring prediction probabilities as a part of input and are only designed for classification tasks, our algorithm is flexible to be employed in various tasks since it takes prediction losses as input. To illustrate its benefits of extending to other tasks, we conduct a case study on BDD100K (Yu et al. 2020), a large-scale dataset composed of driving scenes with abundant annotations. It includes ten tasks, of which we investigate object detection in our paper. The number of images in its object detection task is 79,863, which we split into train, validation, and test datasets with the ratio 2:1:1. We train a YOLOv7 (Wang, Bochkovskiy, and Liao 2023) and try to identify coherent error slices for it. We employ Average Precision (AP) as the performance metric that is widely adopted in detection tasks. Of the 13 categories in the task, we select 2 categories with a relatively high overall performance and a large sample size, i.e. pedestrian and traffic light. We apply our algorithm MCSD for each of them respectively. Note that we do not compare with Domino or PlaneSpot since neither of them is applicable to tasks other than classification.

Table 4 shows that MCSD successfully identifies error slices whose AP are lower than those of overall for both categories, and whose coherence is higher than that of Spotlight in terms of manifold compactness. In Figure 6, each row corresponds to five images randomly sampled from a given source. Left three columns correspond to the category of pedestrians, while right three columns correspond to the category of traffic lights. For both pedestrians and traffic lights, samples from the source of MCSD are coherent in that they are all taken at night. This conforms to the intuition that it is more difficult to recognize and locate objects when the light is poor. However, directly sampling from the high-loss images can hardly exhibit any common patterns. This reveals the potential of MCSD to extend to other tasks.

Toxic?	Yes		No	
Method	Acc. (%) ↓	Comp.↑	Acc. (%) ↓	Comp.↑
Spotlight	48.6	5.10	91.0	7.33
Domino	56.1	5.98	87.9	6.55
PlaneSpot	46.3	1.65	96.5	2.99
MCSD	**25.2**	**8.56**	**60.8**	**7.67**
Overall	61.2	-	90.9	-

Table 5: Results on CivilComments, along with overall accuracy of the trained model. "Comp." means "Manifold Compactness". We mark the best method in bold type. "%" indicates that the digits are percentage values.

Case Study: CivilComments

In addition to experiments on visual tasks, to demonstrate the applicability of our method to other types of data, we conduct experiments on CivilComments (Borkan et al. 2019), a text dataset of 244,436 comments included in popular distribution shift benchmarks (Yang et al. 2023; Koh et al. 2021). Its task is to predict whether a given comment is toxic or not. We follow the data split and training process of Yang et al. (2023) to train a $\text{BERT}_{\text{base}}$. We apply algorithms to toxic and non-toxic comments respectively. In Table 5, we can see that MCSD identifies slices of the lowest accuracy and highest manifold compactness in both categories. We also list two parts of comments that are respectively sampled from the slice identified by applying MCSD to the "toxic" category and from all comments of "toxic" category in Appendix (***Warning***: many of these comments are severely offensive or sensitive), where each part contains 10 comments. We employ ChatGPT to tell the main difference between the two parts of comments and the reply is "Part 1 is characterized by detailed, historical, and ethical discussions with a critical stance on conservatism and a defense of marginalized groups". We further check and confirm that comments in part 1, i.e. the slice identified by our method, mostly present a positive attitude towards minority groups in terms of gender, race, or religion. This implies that the model tends to treat comments with excessively positive attitudes towards minority groups as non-toxic, some of which are actually toxic. These results demonstrate our method's usefulness in text data.

Conclusion

In this paper, inspired by the data geometry property, we propose manifold compactness as a metric of slice coherence without using predefined slice labels. Via explicit metrics, we develop an algorithm that directly incorporates risk and coherence into the optimization objective. We conduct comprehensive experiments to demonstrate both the validity of our proposed metric and the superiority of our algorithm.

Acknowledgements

This work was supported by Tsinghua-Toyota Joint Research Fund, NSFC (No. 62425206, 62141607), and Beijing Municipal Science and Technology Project (No. Z241100004224009). Peng Cui is the corresponding author. All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Belkin, M.; and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6): 1373–1396.

Borkan, D.; Dixon, L.; Sorensen, J.; Thain, N.; and Vasserman, L. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, 491–500.

Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2024. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dann, E.; Henderson, N. C.; Teichmann, S. A.; Morgan, M. D.; and Marioni, J. C. 2022. Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature Biotechnology*, 40(2): 245–253.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

d'Eon, G.; d'Eon, J.; Wright, J. R.; and Leyton-Brown, K. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1962–1981.

Eyuboglu, S.; Varma, M.; Saab, K. K.; Delbrouck, J.-B.; Lee-Messer, C.; Dunnmon, J.; Zou, J.; and Re, C. 2022. Domino: Discovering Systematic Errors with Cross-Modal Embeddings. In *International Conference on Learning Representations*.

Gurobi Optimization, L. 2021. Gurobi optimizer reference manual.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Irvin, J.; Rajpurkar, P.; Ko, M.; Yu, Y.; Ciurea-Ilcus, S.; Chute, C.; Marklund, H.; Haghgoo, B.; Ball, R.; Shpanskaya, K.; et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 590–597.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, 5637–5664. PMLR.

Liu, E. Z.; Haghgoo, B.; Chen, A. S.; Raghunathan, A.; Koh, P. W.; Sagawa, S.; Liang, P.; and Finn, C. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, 6781–6792. PMLR.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36: 34892–34916.

Liu, J.; Wang, T.; Cui, P.; and Namkoong, H. 2023b. On the need for a language describing distribution shifts: Illustrations on tabular datasets. *Advances in Neural Information Processing Systems*, 36.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Melas-Kyriazi, L. 2020. The mathematical foundations of manifold learning. *arXiv preprint arXiv:2011.01307*.

Pedronette, D. C. G.; Gonçalves, F. M. F.; and Guilherme, I. R. 2018. Unsupervised manifold learning through reciprocal kNN graph and Connected Components for image retrieval tasks. *Pattern Recognition*, 75: 161–174.

Plumb, G.; Johnson, N.; Cabrera, A.; and Talwalkar, A. 2023. Towards a More Rigorous Science of Blindspot Discovery in Image Classification Models. *Transactions on Machine Learning Research*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500): 2319–2323.

Tian, Y.; Ye, Q.; and Doermann, D. 2025. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475.

Wang, F.; Adebayo, J.; Tan, S.; Garcia-Olano, D.; and Kokhlikyan, N. 2023. Error Discovery by Clustering Influence Embeddings. *Advances in Neural Information Processing Systems*, 36.

Yang, Y.; Zhang, H.; Gichoya, J. W.; Katabi, D.; and Ghassemi, M. 2024. The limits of fair medical imaging AI in real-world generalization. *Nature Medicine*, 30(10): 2838–2848.

Yang, Y.; Zhang, H.; Katabi, D.; and Ghassemi, M. 2023. Change is Hard: A Closer Look at Subpopulation Shift. In *International Conference on Machine Learning*, 39584–39622. PMLR.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zemel, R.; and Carreira-Perpiñán, M. 2004. Proximity graphs for clustering and manifold learning. *Advances in neural information processing systems*, 17.