Leveraging Data Symmetries to Select an Optimal Subset of Training Data under Label Noise

Kumar Shubham
Indian Institute of Science, Bengaluru, India
shubhamkuma3@iisc.ac.in

Pavan Karjol
Indian Institute of Science, Bengaluru, India
pavankarjol@iisc.ac.in

Kiran M. K.
Indian Institute of Science, Bengaluru, India
kiranmk@iisc.ac.in

Prathosh A. P.
Indian Institute of Science, Bengaluru, India
LatentForce.ai
prathosh@iisc.ac.in

Abstract

The performance of machine learning models often relies on large labeled datasets; however, data collected from diverse sources can contain label noise. Recent work has shown that, in noisy settings, there may exist a subset of the training data on which models can achieve performance comparable to training on a noise-free dataset. A widely used method for identifying such subsets is *cutstats*, which employs k-nearest neighbors (k-NN) to detect low-noise samples. However, its performance on high-dimensional data remains largely unexplored. In this work, we formally establish that the performance of a classifier trained on a subset of a noisy dataset selected via *cutstats* is influenced by the accuracy of k-NN. We further demonstrate that, in noisy environments, exploiting data invariance and knowledge of underlying symmetries can significantly enhance the performance of k-NN, bringing it closer to the Bayes optimal classifier even in high-dimensional regimes. Finally, we show that for real-world scenarios, where information about the underlying invariance is only partially known, learnt invariant representations can still facilitate the identification of near-optimal subsets. Our code is available at https://github.com/kyrs/group-invariant-subset-selection.

1 Introduction

In modern AI and scientific applications, both data quality and quantity are crucial for training reliable downstream models and supporting rigorous empirical analysis. High-quality datasets are not only essential for model development but also carry substantial economic and proprietary value [1–3]. However, data aggregated from heterogeneous sources can contain label noise, which may degrade model performance and compromise the reliability of subsequent analyses.

Several methods have been proposed to mitigate the effect of noisy labels by improving model robustness during training [4–12]. These approaches, however, primarily address robustness at the model level rather than improving the quality of the training data itself. Recent work [13] shows that selecting high-quality subsets from noisy datasets can improve downstream classifier performance while also supporting deeper scientific analysis, particularly in domains such as cancer research. For instance, carefully curated subsets of patient genomic data can improve drug recommendation and reveal meaningful biological patterns, which is especially valuable, when acquiring new labeled data is costly or limited. Similar benefits have been reported across other scientific domains [14–16], highlighting the broader importance of effective subset selection.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Recently, Lang et al. [17] demonstrated that, even in the presence of label noise, there exists an subset of the training data on which models can achieve performance comparable to that obtained using noiseless data, under appropriate concentration bounds. To identify such subsets, they proposed using *cutstats* [18], a k-nearest-neighbor (k-NN)-based technique that selects candidate samples by comparing each sample's features and label with those of its local neighborhood. Since *cutstats* relies only on the training data and its induced feature space, it does not require prior knowledge of the downstream classifier. This makes it especially useful when the classifier to be trained on the selected subset is not known in advance.

However, despite its promise, the effectiveness of *cutstats* on high-dimensional data, and its ability to recover an optimal subsets consistent with a Bayes-optimal selection rule remain largely unexplored [17]. In this work, we investigate the performance gap between classifiers trained on noisy subsets and those trained on noiseless datasets in high-dimensional settings.

We show that, for *cutstats*, this gap is determined by the accuracy of k-NN in identifying noiseless samples. In low-dimensional regimes, k-NN can approximate the Bayes classifier[1] (the theoretically optimal classifier) and thereby recover optimal subset i.e., the subset selected by bayes optimal classifier. In high-dimensional regimes, however, this approximation may not hold, and hence can lead to the selection of suboptimal subsets.

To address this limitation, we show that incorporating prior knowledge of symmetries and their associated invariances can preserve the effectiveness of k-NN in high-dimensional settings. Invariances such as rotation and permutation commonly arise in scientific datasets [20, 21], where they capture structured properties of the data that can be leveraged to distinguish clean samples from noisy ones. By constructing representations that explicitly encode these invariances, we theoretically show that the approximation to the Bayes classifier is preserved under orthogonal and permutation-based symmetries, thereby enabling the recovery of optimal subsets even in high-dimensional regimes. Finally, to account for practical settings in which the exact invariant representation is unknown, we further show that invariant representations like group-invariant representation learning [22] and contrastive learning [23] can still improve the performance of *cutstats*.

Our contribution can be summarized as follows:

- We show that the performance gap of a classifier trained on a subset of the training data selected using *cutstats* is governed by the accuracy of k-nearest neighbors (k-NN) under label noise. Our analysis further shows that k-NN can approximate the Bayes classifier in such settings, thereby enabling optimal subset selection.
- We provide a theoretical analysis of how this approximation with the Bayes classifier breaks down in high-dimensional regimes, where optimal subset selection is no longer guaranteed.
- We demonstrate that exploiting symmetries and associated invariances, specifically orthogonal and permutation symmetries, can restore this approximation and ensure the recovery of optimal subsets.
- For practical settings, we demonstrate that invariant representations learned through group-equivariant networks or contrastive learning can significantly enhance the performance of *cutstats*, enabling the selection of accurate subsets of the dataset.

2 Related Work

2.1 Handling Label Noise

The effect of label noise is often mitigated by training robust classifiers. A common strategy to achieve this is by using noise-aware loss functions [4, 24, 6, 11], which reduce overfitting to noisy labels and improve generalization. However, these methods can be difficult to optimize, often require extensive hyperparameter tuning, and do not address the quality or proprietary value of the dataset itself.

Unlike loss function-based techniques, recent approaches [25, 26] aim to improve model robustness not only by incorporating loss-based strategies but by also using appropriate samples for train-

[1] This equivalence with the Bayes classifier holds for specific noise rates and with certain probability, as shown by Bahri et al. [19].

ing. Many of these methods [27–30] adopt a co-training strategy that seeks to identify appropriate samples to improve the robustness of the model. On a similar line, self-supervised learning(SSL) techniques improve the performance of the model [31] using relabeling via co-teaching [32], and selecting or filtering samples during training [33, 34]. However, such approaches often require training multiple networks, which leads to increased memory usage and higher computational costs.

To address these limitations, several methods have been proposed to select a clean subset of the training dataset. Recent work has shown that, in the presence of noise, a representative subset of data points [35] or samples with low predictive uncertainty [17] can be selected from the training set. Similarly, identifying samples that are frequently forgotten during training can also be used to remove detrimental noisy samples [36], thereby enabling the selection of high-quality subsets of the training dataset. Among these approaches, *cutstats* [17] stands out for its simplicity, as it selects subsets of the training data solely based on input features and their associated labels, without any dependency on the downstream classifier. Such methods are particularly relevant in enterprise and crowdsourcing settings [37], where datasets are curated without prior knowledge of the downstream models that will be trained on them. In these scenarios, classifier-agnostic subset selection is essential to ensure broad applicability and robustness across diverse learning tasks. However, despite its practical advantages, the effect of data dimensionality on the performance of *cutstats* remains largely unexplored.

2.2 Classifier Performance and Group Invariance

Real-world datasets often exhibit inherent symmetries, where the underlying properties of the data remain unchanged under transformations such as translation, rotation, and permutation [21, 38]. These symmetries naturally arise from the physical or structural properties of data in domains such as molecular modeling, protein structures, crystalline materials, point clouds, and set-valued data [14–16]. Exploiting such symmetries can significantly improve the performance of machine learning models, particularly when structured patterns are present. Recent advances in equivariant neural networks have demonstrated that incorporating rotational and permutation symmetries into model architectures can enhance predictive performance [22]. However, many of these applications also involve intrinsic label noise, costly clean annotations, and the need for curated datasets that can be reused across downstream scientific analyses. Motivated by these challenges, we propose a data-centric subset-selection method that leverages known symmetries to identify reliable training samples in the presence of label noise. Our work provides an initial step toward understanding how symmetry influences subset selection and offers theoretical insights to guide the development of unsupervised invariant representations for this task.

3 Methods

3.1 Problem Definition

Let the noiseless dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$, where each input $x_i \in \mathbb{R}^d$ is a d-dimensional feature vector drawn from an underlying distribution $\mathcal{P}$, and the corresponding label (y_i) is generated by true-labeling function $\boldsymbol{f} : \mathbb{R}^d \to \{0, 1\}$ such that $y_i = \boldsymbol{f}(x_i)$. Suppose the data admits an underlying symmetry, represented by a group G such that class labels are invariant under the group action, i.e.,: $\boldsymbol{f}(g \cdot x) = \boldsymbol{f}(x) \quad \forall g \in G$, where $(\cdot)$ denotes group action.

For a group G, the orbit of any input x is defined as $\mathcal{O}_x = \{g \cdot x : g \in G\}$. Let $\mathcal{F}$ be an invariant representation function associated with G, such that all points in the same orbit share the same representation: $\mathcal{F}(x) = \mathcal{F}(x') \quad$ whenever $x' \in \mathcal{O}_x$.

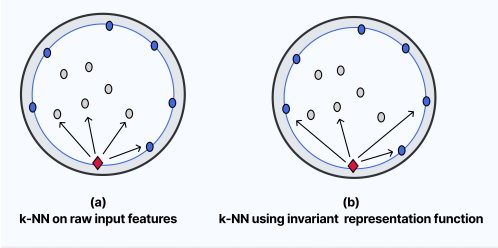

Figure 1: Nearest-neighbor selection under orthogonal group invariance. Points along the circle's circumference belong to the same orbit and may share class labels. (a) Standard k-NN may select neighbors outside the orbit (Section 3.1). (b) Invariant representations enable neighbor selection from within the orbit.

Moreover, $\mathcal{F}$ retains necessary class-relevant information [39, 40]. In practice, the true labeling function $\boldsymbol{f}$ is not observable, and instead we have access to noisy approximations of the labels. Let $\mathcal{D}_{\text{noisy}} = \{(x_i, \hat{y}_i) \mid i = 1, \ldots, m\}$, where $\hat{y}_i$ denotes a noisy version of the ground-truth label.

The goal of this work is to identify a subset $\mathcal{S} \subseteq \mathcal{D}_{\text{noisy}}$ using the *cutstats* procedure, while leveraging the invariant representation function $\mathcal{F}$, such that a classification model $\mathcal{M}$ trained on $\mathcal{S}$ approximates the true labeling function($\boldsymbol{f}$).

3.2 Methodology

To achieve this objective, we first theoretically analyze the error gap of a model trained using a subset of a noisy dataset with its noiseless counterpart. For practical scenarios where the noiseless datasets are not available, we show that the resulting error gap for *cutstats*, depends on the accuracy of k-NN in identifying correct samples from the noisy dataset. In low-dimensional settings, it can closely approximate the Bayes-optimal classifier and, consequently, achieve near-optimal classification performance. However, this approximation might not hold as the dimensionality increases.

We then demonstrate that incorporating the invariant representation $\mathcal{F}$ into subset selection can restore this approximation in high-dimensional regimes, particularly for orthogonal and permutation groups. In practical scenarios where the underlying invariances are unknown or only partially specified, we show that group-invariant representation learning or contrastive learning can provide effective approximations.

In the following section, we formally discuss the *cutstats* technique in detail and describe how knowledge of $\mathcal{F}$ can be integrated into the subset selection process.

3.3 Cutstats

Cutstats provides an efficient technique for selecting a subset of the noisy training dataset ($\mathcal{D}_{noisy}$) based on the agreement between the noisy class label ($\hat{y}_i$) of a sample (x_i) and the labels of its nearest neighbors NN(x_i). In *cutstats*, the selection of the subset $\mathcal{S} \subseteq \mathcal{D}_{noisy}$ is based on the Z-score (z_i), defined as:

$$
\begin{aligned}
z_i &= \frac{J_i - \mu_i}{\sigma_i}, & J_i &= \sum_{j \in \mathcal{N}_i} w_{ij}\, \mathbf{1}(\hat{y}_j \neq \hat{y}_i)\,, \\
w_{ij} &= (1 + \|x_i - x_j\|_2)^{-1}\,, & \mu_i &= (1 - p_i) \sum_{j \in \mathcal{N}_i} w_{ij}, \\
\sigma_i^2 &= p_i(1 - p_i) \sum_{j \in \mathcal{N}_i} w_{ij}^2, & \mathcal{N}_i &= \mathrm{NN}(x_i), \quad p_i = \mathbb{P}(\hat{y}_i).
\end{aligned}
\tag{1}
$$

where the weights $w_{i,j}$ capture feature similarity between x_i and its neighbors $x_j \in \text{NN}(x_i)$. The score J_i measures the weighted disagreement with neighbors' labels, while μ_i and σ_i^2 denote the expected mean and variance of J_i under the null hypothesis of independent label assignment, estimated using the label distribution $\big(\mathbb{P}(\hat{y})\big)$ in the training dataset $\mathcal{D}_{noisy}$.

Small z_i values indicate strong agreement between a sample and its neighborhood, whereas large z_i values suggest disagreement between a sample's label and that of its nearest neighbor, signaling potential label noise.

Intutively, *cutstats* select the datset by comparing the label of each sample with those of its local neighborhood. Samples whose labels show high disagreement with their neighbors, are treated as likely noisy candidates for removal. Based on this disagreement score captured by (z_i), *cutstats* ranks training samples and select the top $\tau\%$ with the smallest values of z_i as $\mathcal{S}$, and discard remaining samples.

While the classical *cutstats* uses raw input features [17] to define weights and nearest neighbors. The invariant representation function ($\mathcal{F}$) can be incorporated for calculating the weights and identifying the nearest neighbor as $w_{i,j} = \big(1 + \|\mathcal{F}(x_i) - \mathcal{F}(x_j)\|_2\big)^{-1}$, where the neighborhood is now defined as $NN(x_i) = \{\, x_j : x_j$ is a nearest neighbor of x_i with respect to the ℓ_2 distance in the representation space $\mathcal{F}\}$.

Once a subset of the noisy training dataset has been selected, a downstream classifier ($\mathcal{M}$) is trained on it and subsequently used to predict the true class labels of new, unseen samples.

Since *cutstats* assigns smaller z-values when the majority of neighbors share the same class as the selected data point, our analysis will primarily focus on the nearest neighbors and their labels. We will next analyze the performance of the classifier trained on the selected subset of the noisy dataset.

3.4 Performance of Selected Subset and Noiseless Dataset

The generalization gap between a classifier trained on a subset of noisy data and those trained on the noiseless data has been formally characterized by Lang et al. [17]. In their framework, a subset selection heuristic (e.g., *cutstats*) may abstain from including a training sample $(x, \hat{y})$ in subset ($\mathcal{S}$) by assigning it the special label ϕ (i.e., $\hat{y} = \phi$, where $\hat{y} \in \{0, 1, \phi\}$). In such a setting, Lang et al. [17] formally present a bound on the performance of a classifier using balanced error (err_{bal}), defined as :

$$err_{bal}(\mathcal{M}, y) = \frac{1}{2}\Big(\mathbb{P}(\mathcal{M}(x) = 1 \mid y = 0) + \mathbb{P}(\mathcal{M}(x) = 0 \mid y = 1)\Big)$$

For a classifier $\mathcal{M}$ trained on noisy labels $\hat{y}$, this balanced error can be compared against that of an optimal classifier $\mathcal{M}^*$ trained on noiseless data. The performance gap largely depend upon the non-abstention rate $\mathbb{P}(\hat{y} \neq \phi)$ and the noise rates of the selected subset (α, γ) for $\hat{y} = 1$ and $\hat{y} = 0$ respectively. The complete formulation is provided below :

Lemma 3.1. *[17] Under the assumptions outlined by Lang et al. [17]. Let Γ denote the hypothesis class of the downstream model used for training the classifier. Furthermore, let $\mathcal{M}^*$ represent the optimal classifier trained on noiseless class labels (y), defined as*

$$\mathcal{M}^* = \inf_{\mathcal{M} \in \Gamma} \textit{err}_{\textit{bal}}(\mathcal{M}, y),$$

and $\mathcal{M}$ denote the classifier obtained by minimizing the empirical balanced error on the noisy dataset ($\mathcal{D}_{noisy}$) with m training samples. Then, with probability $1 - \delta$,

$$\begin{aligned} &\textit{err}_{\textit{bal}}(\mathcal{M}, y) - \textit{err}_{\textit{bal}}(\mathcal{M}^*, y) \\ &\leq \tilde{O}\left(\frac{1}{1 - \alpha - \gamma}\sqrt{\frac{\textit{VC}(\Gamma) + \log\frac{1}{\delta}}{m\mathbb{P}[\hat{y} \neq \phi] \min_y \mathbb{P}[\hat{y} = y \mid \hat{y} \neq \phi]}}\right), \end{aligned} \tag{2}$$

where $\tilde{O}$ hides logarithmic factors in m, $\textit{VC}(\Gamma)$, and

$$\alpha = \mathbb{P}(y = 0|\hat{y} = 1), \gamma = \mathbb{P}(y = 1|\hat{y} = 0)$$

denoted the error rates in the selected subset.

As stated in Lemma 3.1, the performance of any downstream model trained on a noisy subset of data depends on two primary factors: (i) the level of noise present in the selected set (α, γ), and (ii) the effective size of the dataset used for training, $\big(m\mathbb{P}[\hat{y} \neq \phi]\big)$.

For any given noisy dataset with m training samples ($\mathbb{P}(\hat{y} \neq \phi) = 1$) and the error rates $\alpha_{\text{noisy}}, \gamma_{\text{noisy}}$, an appropriate subset can be selected by abstaining from the noisy samples. This results into generation of a new dataset $\mathcal{S}$ with error rates defined as $\alpha_{\mathcal{S}}, \gamma_{\mathcal{S}}$. For k-NN based methods, like *cutstats*, the formal relationship between the error rates of the original data and the subset can be characterized as follows:

Proposition 3.2. *[Performance of k-NN] Consider the setup of Lemma 3.1. for k-NN ($\mathcal{H} : x \mapsto y^{knn}$), with y^{knn} as the class label of the majority of the samples in the nearest neighbor and*

$$\lambda_0 = \mathbb{P}(y^{knn} = 0 \mid y = 0), \qquad \lambda_1 = \mathbb{P}(y^{knn} = 1 \mid y = 1)$$

denote the class-wise accuracies of k-NN for class 0 and class 1, respectively. Then the error rate of the subset selected by k-NN satisfies:

$$\alpha_{\mathcal{S}} = \frac{\alpha_{noisy}(1 - \lambda_0)}{\alpha_{noisy}(1 - \lambda_0) + (1 - \alpha_{noisy})\lambda_1}, \qquad \gamma_{\mathcal{S}} = \frac{\gamma_{noisy}(1 - \lambda_1)}{\gamma_{noisy}(1 - \lambda_1) + (1 - \gamma_{noisy})\lambda_0}$$

Proofs of the propositions and corollaries are provided in the Appendix F.1.

Intuitively, more accurate heuristics (with λ_0, λ_1 close to 1) introduce less noise during subset selection. This relationship is formally described below:

Corollary 3.3. *Suppose the class priors are balanced, i.e.,* $\mathbb{P}(y=0) = \mathbb{P}(y=1) = \frac{1}{2}$. *If the k-NN classifier has error* $\mathbb{P}(y^{knn} \neq y) \leq \frac{1}{2}$, *then the subset error rates satisfy*

$$\alpha_{\mathcal{S}} \leq \alpha_{noisy} \quad \textit{and} \quad \gamma_{\mathcal{S}} \leq \gamma_{noisy}.$$

Moreover, as $\lambda_0, \lambda_1 \to 1$, *the subset error rates also reduces:*

$$\alpha_{\mathcal{S}},\ \gamma_{\mathcal{S}}\ \to\ 0.$$

Corollary 3.3 and Lemma 3.1 establish a relationship between the precision of k-NN in the presence of label noise and that of a downstream classifier trained on a subset of data selected by it. Since, in theory, the best performance achievable by any classifier corresponds to that of the Bayes classifier, in the next section we analyze the conditions under which k-NN can approximate the Bayes classifier in a noisy setting.

3.5 Bayes optimal classifier and K-NN

The relationship between the performance of k-NN and the Bayes classifier in the presence of label noise has been formally established by Bahri et al. [19]. According to their analysis, if a dataset contains n uncorrupted samples and $\mathcal{C}$ samples with label noise. Then, for k-nearest neighbors classifier ($\eta_k(x)$), and the Bayes-optimal classifier ($\eta^*(x)$) following holds:

Lemma 3.4. *[19] Let* $\nu > 0$, *and assume the conditions of Bahri et al. [19] hold under the Tsybakov noise condition with parameter* β. *Then there exist constants* $K_l(d), K_u(d), K, K' > 0$, *depending only on the dimension* d *and the underlying distribution, such that with probability at least* $1-\nu$ *:*

If k *lies in the range*

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n,$$

Then,

$$\mathbb{P}\big(\eta_k(x) \neq \eta^*(x)\big) \leq K \cdot \lambda^{\beta}, \qquad R_X - R^* \leq K' \cdot \lambda^{\beta+1},$$

where

$$\lambda = \left(\sqrt{\frac{\log n + \log(1/\nu)}{k}} + \left(\frac{k}{n}\right)^{\rho/d} \right),$$

R_X *and* R^* *denote the risks of the k-NN and the Bayes optimal classifier, respectively.*

where $\Delta(\mathcal{C})$ denotes the minimum distance between features of corrupted examples and ρ is a parameter associated with Holder's continuity. Details about all the parameters and the associated assumptions are provided in the Appendix E, all the notations are described in Appendix A.

An important consequence of Lemma 3.4 is that, under noisy conditions , it provides a sufficient condition for the existence of a range of values of k for which the k-NN can perform similar to the Bayes optimal classifier. In such cases, k-NN can approximate an optimal classifier and can therefore identify an optimal subset of data points without incurring the additional computational cost of training a large model for this task. However, in high-dimensional settings, this approximation with the Bayes classifier may not hold.

In fact, when the feature dimension d is large, the sufficient condition fails to hold for any value of k, and the approximation to the Bayes-optimal classifier may no longer be valid.

Proposition 3.5. *[High-dimensional* k*-NN and Bayes Classifier] As the feature dimension* d *increases, the factor* $K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$ *in Lemma 3.4 decreases with* d $\big(K_u(d)$ *decreases faster than any fixed exponential base with* $d\ \big)$. *Consequently, there exists a threshold* d_0 *such that for all data dimension* $d \geq d_0$ *and for a given n,* ν, ρ *and* Δ *no* k *satisfies*

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n.$$

Proof of the given proposition is provided in Appendix F.3.

As per the given Proposition 3.5, the k-NN model may not approximate the performance of the Bayes optimal classifier for high-dimensional data. This limitation reduces the efficacy of k-NN in high-dimensional settings, as it may not guarantee performance comparable to the Bayes optimal classifier when no such k exists.

3.6 Orthogonal and Permutation Group

To address this problem in high-dimensional settings, we show that for datasets whose labeling function ($\boldsymbol{f}$) is invariant under the $\mathrm{SO}(d)$ or S_d group (Section 3.1), incorporating an invariant representation function ($\mathcal{F}$) into the nearest-neighbor classifier (Section 3.3) yields performance that remains approximately as good as the Bayes-optimal classifier and recover the sufficient condition in high dimensions.

Proposition 3.6. *[Orthogonal Group] For data whose true-labeling function satisfy orthogonal invariance ($SO(d)$) as per section 3.1, if k-NN utilizes the invariant representation function ($\mathcal{F}$), and the inequality holds for dimension (d=1), i.e.,*

$$K_l(1) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+1}} \leq k \leq K_u(1) \cdot \Delta(\mathcal{C}) \cdot n.$$

Then there exists a k for all dimensions d for which k-NN will approximate Bayes optimal classifier for a given n, ν, ρ and Δ,

Proposition 3.7. *[Permutation Group] For data whose true-labeling function satisfy permutation invariance (S_d) as per section 3.1, if k-NN utilizes the invariant representation function ($\mathcal{F}$), then, $K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$ does not decrease with d, and there will exist k for a given n, ν, ρ and Δ for which*

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n.$$

holds, and k-NN can approximate Bayes optimal classifier.

Proof of the given propositions is provided in Appendix F.3. For high-dimensional data, even though the desired approximation of the Bayes optimal classifier (Lemma 3.4) might not be achieved directly from the input features x, as established by Proposition 3.5, Propositions 3.6 and 3.7 demonstrate that this approximation can be recovered by employing an invariant representation function $\mathcal{F}$, which enables the selection of informative subsets of the training data for downstream classifier training.

3.7 Invariant Representation Function

A key requirement of the proposed approach is access to an invariant representation function $\mathcal{F}$, as discussed in Section 3.1, which preserves necessary class-relevant information and therefore yields the same bayes error as the original input features. However, in many real-world scenarios, the exact invariant representation is not known and should be approximated from data. Such representations are commonly obtained using methods including unsupervised group-invariant representation learning [22] and contrastive learning [23], which aim to capture the intrinsic invariances present in the data without relying on ground-truth labels. While these representations can facilitate the efficient utilization of the underlying dataset, they may not capture the underlying class-relevant information entirely, resulting in higher Bayes error [41, 42]. Nevertheless, in practice [43], such representations often yield strong downstream performance. This is likely because they suppress noise and capture underlying invariances in the data. While such representations facilitate more efficient utilization of the dataset, their performance is governed by how faithfully they retain discriminative information necessary for accurate classification [41, 42].

4 Experiments

4.1 Settings

In this work, we conduct experiments to analyze four key aspects. First, we investigate whether knowledge of the underlying invariant function ($\mathcal{F}$) improves the performance of *cutstats* when the

Table 1: Details about the group, associated invariant function, and generating function used for the experiment. we have used norm operator as invariant function for Orthogonal group and sort function for Permutation group where $\text{sort}(\{f_1, \ldots, f_n\}) = (f_{(1)}, \ldots, f_{(n)})$ where $f_{(1)} \leq \cdots \leq f_{(n)}$.

Group name	Invariant function	Generating function
Orthogonal Group	$\mathcal{F}(x) \to \|\|x\|\|_2$	$h(x) = k_1 \sin(c_1 x^\top x) + k_2 \sin^2(c_2 x^\top x) + k_3 \cos(c_3 x^\top x)$
Permutation Group	$\mathcal{F}(x) \to \text{sort}(x)$	$h(x) = \sum_{k=1}^{l} \sum_{i=1}^{d} \sin(x_i^k)$

dataset satisfies orthogonal and permutation invariance. For this experiment, we use synthetic data and the invariant functions outlined in Table 1. Second, we conduct experiments on real-world datasets like rotated MNIST [44] where the labeling function satisfies $\text{SO}(2)$ invariance, but the underlying invariant function is not known and must be learned from the data. To further analyze the influence of the Bayes error, we consider two methods: (i) unsupervised group invariant representation learning [22], which leverages the knowledge about underlying symmetries and associated group to learn an invariant representation function **ours (group inv)**, and (ii) contrastive learning [23], which learns representations solely based on data augmentations **ours (contrastive)**. third, for scenarios where underlying symmetries are not known a priori, we compare the contrastive version of invariant representation learning on human-generated label noise in CIFAR10N [45]. Finally, we analyze how errors in the learned representation influence the ability of our method to identify accurate subsets of the training dataset. Additional experiments on the impact of dimension on the performance of K-NN, comparison with robustness-based baselines, asymmetric noise and performance on dataset(tetris) with other symmetries are provided in Appendix K, I, J, G, respectively.

For experiments with synthetic datasets, we select functions whose classes are invariant to orthogonal and permutation group actions. We generate inputs x sampled from a uniform distribution and assign class labels by thresholding $h(x)$, as defined in Table 1. Since the function $h(x)$ is invariant under group transformations, the resulting true class labels y also remain invariant. To introduce label noise, we randomly flip class labels in the training data with probability p. Further experiment on asymmetric noise is provided in Appendix J. As a downstream classifier, we use a two-layer neural network for the synthetic dataset, experiments related to rotated MNIST and CIFAR-10N use Resnet-9 architectures, experiments related to Tetris (Appendix G) use GNN. Further details on the data generation process and training specifications are provided in the Appendix B. We compare our approach against the following baselines: **NoiseLess**: the original dataset without label noise. **Noisy-Complete**: the complete noisy dataset with the same features but corrupted labels, **Random**: a random subset of the noisy dataset, **Entropy**: a setting where high-entropy examples are excluded from training [17], **Vanilla cutstats**: *cutstats* using raw input features for subset selection [17], **Forget**: a setting where frequently forgotten samples are removed during training [36], **Herding**: a setting where only representative samples of the dataset are retained [35]. We primarily consider subset-selection baselines because, like our approach, they produce curated datasets that can be reused independently of the downstream classifier. Moreover, methods such as Entropy, Forgetting, and CutStats have previously been employed to analyze label noise in training data, making them particularly relevant for comparison. All subset selection methods utilize the same proportion of the data, (40% of the noisy training dataset following [17]), with the only difference being the specific set of points included in the subset (see Appendix M,N for ablation on subset size and nearest neighbor). We report both the accuracy of the downstream classifier (**Classifier-Acc**) and the percentage of correctly labeled samples in the selected subset (**Subset-Acc**).

4.2 Orthogonal and Permutation Group

Table 2 presents a comparison of baseliens and our method for orthogonal and permutation groups. Since the dataset corresponds to a binary classification problem with only two classes, we use a high noise probability ($p = 0.45$) for this experiment. As per the results,"NoiseLess" baseline serves as an upper bound with clean data for both classifier and subset accuracy. In contrast, the "Noisy-Complete" baseline highlights the detrimental impact of label noise on performance. Among all subset selection methods, our proposed approach ("Ours") achieves the best performance, with classifier accuracies of 68.52% and 86.00% for the orthogonal and permutation groups, respectively. It

Table 2: Comparison of downstream classifier accuracy and subset accuracy across different methods. Results are reported as mean across three different seeds, with the best subset selection method highlighted in **bold**.

Method	Orthogonal Group		Permutation Group	
	classifier-Acc	Subset-Acc	classifier-Acc	Subset-Acc
NoiseLess	70.40 ± 0.83	100.00 ± 0.00	90.93 ± 0.30	100.00 ± 0.00
Noisy-Complete	50.83 ± 8.11	55.08 ± 0.11	78.73 ± 3.45	55.33 ± 0.82
Random	45.72 ± 3.26	54.93 ± 0.57	77.33 ± 4.77	55.50 ± 0.49
Vanilla Cutstats	63.63 ± 4.81	60.51 ± 1.58	82.70 ± 4.37	71.21 ± 3.09
Entropy	62.53 ± 6.64	56.46 ± 0.33	70.72 ± 4.61	56.05 ± 1.36
Forget	68.45 ± 0.53	58.69 ± 1.13	75.98 ± 2.88	63.04 ± 1.96
Herding	51.32 ± 0.98	55.66 ± 0.51	71.00 ± 2.35	54.13 ± 0.77
Ours	$\mathbf{68.52 \pm 0.66}$	$\mathbf{74.31 \pm 1.10}$	$\mathbf{86.00 \pm 0.84}$	$\mathbf{74.42 \pm 3.15}$

substantially outperforms competing methods by up to 3.3% in classifier accuracy for the permutation group and 13.8% in subset accuracy for the orthogonal group, demonstrating its effectiveness in identifying clean, reliable subsets from noisy data.

4.3 Experiment on Rotated MNIST

Table 3: Performance comparison on Rotated MNIST dataset under different noise probabilities.

Method	Noise 0.4		Noise 0.6		Noise 0.8	
	Classifier-Acc	Subset-Acc	Classifier-Acc	Subset-Acc	Classifier-Acc	Subset-Acc
NoiseLess	96.80 ± 0.27	100.00 ± 0.00	96.80 ± 0.27	100.00 ± 0.00	96.80 ± 0.27	100.00 ± 0.00
Noisy-Complete	86.63 ± 1.13	59.98 ± 0.28	79.51 ± 8.13	39.94 ± 0.09	60.27 ± 3.49	20.07 ± 0.16
Random	87.68 ± 2.08	59.99 ± 0.34	60.98 ± 0.42	39.88 ± 0.25	37.60 ± 8.52	20.07 ± 0.33
Vanilla Cutstats	90.86 ± 1.77	99.65 ± 0.06	83.78 ± 6.52	86.51 ± 0.08	68.93 ± 2.87	35.33 ± 0.45
Entropy	80.88 ± 5.96	72.64 ± 0.79	69.79 ± 1.54	47.18 ± 0.69	36.59 ± 2.90	21.85 ± 0.28
Forget	65.89 ± 0.14	54.95 ± 0.51	42.63 ± 3.27	28.19 ± 1.26	2.93 ± 1.44	8.51 ± 0.67
Herding	87.01 ± 1.82	62.51 ± 0.52	79.56 ± 4.17	41.08 ± 0.39	39.93 ± 3.11	20.24 ± 0.10
Ours (contrastive)	83.95 ± 2.04	78.16 ± 0.18	76.28 ± 0.57	53.30 ± 0.12	41.52 ± 3.71	24.24 ± 0.20
Ours (group inv)	$\mathbf{91.81 \pm 0.58}$	$\mathbf{99.82 \pm 0.02}$	$\mathbf{89.37 \pm 1.83}$	$\mathbf{90.51 \pm 0.44}$	$\mathbf{75.73 \pm 5.99}$	$\mathbf{37.77 \pm 0.41}$

Table 3 presents a comparison of the baselines on the rotated MNIST dataset across three different noise levels. The invariant group representation ("Ours (group-inv)") consistently achieves the best downstream classifier accuracy (91.81%, 89.37%, and 75.73%, respectively) and subset accuracy (99.82%, 90.51%, and 37.77%) across all noise levels, demonstrating superior robustness. In contrast, under the same experimental setup, the contrastive learning method can generate higher Bayes error as it learns representation using weak signals associated with data augmentations, and specific knowledge of symmetries can be efficiently encoded using group equivariant networks as in ("Ours (group-inv)")[23, 46].

4.4 Experiment on CIFAR-10N

We have further evaluated **Ours (contrastive)**, on the worst-label annotation of **CIFAR-10N** to replicate the setting where the underlying symmetries are not explicitly known. As per the result in Table 4, our method outperforms all baselines across both classifier accuracy and subset accuracy.

Table 4: Classifier accuracy and subset accuracy on CIFAR-10N (Worst Labels).

Method	Classifier-Acc	Subset-Acc
NoiseLess	82.59 ± 0.32	100.00 ± 0.00
Noisy-complete	58.55 ± 2.68	59.79 ± 0.00
Random	58.78 ± 0.27	60.02 ± 0.34
Vanilla Cutstats	51.42 ± 1.53	77.25 ± 0.00
Entropy	62.17 ± 2.80	78.15 ± 0.54
Forget	50.54 ± 1.96	51.73 ± 0.28
Herding	60.08 ± 0.41	62.31 ± 0.25
Ours (contrastive)	$\mathbf{65.30 \pm 1.88}$	$\mathbf{90.48 \pm 0.25}$

4.5 Noisy Representation

To analyze scenarios where the learned representation does not fully capture the underlying symmetry, we conducted an ablation for the orthogonal group. Specifically, we examined how the *invariance error* [47], defined as $\mathbb{E}\left(|h(x) - h(g \cdot x)|\right)$, where g is a group action and $h(x)$ is the underlying generating function, affects the accuracy of the subset selected by k-NN under the same noisy conditions as in Table 2. As shown in Table 5, as the error in the representation increases, the performance of the model in identifying an accurate set decreases.

Table 5: Mean invariance error vs. subset selection accuracy for the Ortho Group.

Mean Invariance Error	Accuracy
0	**74.31 ± 1.10**
0.049	73.86 ± 1.07
0.111	72.52 ± 0.71
0.297	68.53 ± 0.46
0.452	65.06 ± 1.13

5 Conclusion

Training datasets often exhibit label noise, which can substantially degrade downstream model performance. We demonstrate that leveraging underlying symmetries via invariant representations enables principled selection of high-quality subsets from noisy data for classification. Our results further suggest that learned invariant representations remain effective for noise mitigation. The proposed approach, however, depends on the fidelity with which group invariance is captured and the extent to which class-discriminative information is preserved. Future work will extend this analysis to broader classes of groups and data modalities.

6 Acknowledgment

Kumar Shubham is supported by the Kotak IISc AI–ML Centre (KIAC) Fellowship, as well as travel grants from KIAC and Google. Pavan Karjol is supported by the Government of India through the Prime Minister's Research (PMRF) Fellowship.

References

[1] Feng Xiong, Maoyue Xie, Lingjuan Zhao, Cheng Li, and Xuan Fan. Recognition and evaluation of data as intangible assets. *Sage Open*, 12(2):21582440221094600, 2022.

[2] Alberto De Franceschi and Michael Lehmann. Data as tradeable commodity and new measures for their protection. *Italian LJ*, 1:51, 2015.

[3] Herbert Zech. Data as a tradeable commodity–implications for contract law. In *Proceedings of the 18th EIPIN Congress: The New Data Economy between Data Ownership, Privacy and Safeguarding Competition, Edward Elgar Publishing, Forthcoming*, 2017.

[4] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[5] Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*, 2019.

[6] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330, 2019.

[7] Li Yi, Sheng Liu, Qi She, A Ian McLeod, and Boyu Wang. On learning contrastive representations for learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16682–16691, 2022.

[8] Ling-Hao Chen, Yuanshuo Zhang, Taohua Huang, Liangcai Su, Zeyi Lin, Xi Xiao, Xiaobo Xia, and Tongliang Liu. Erase: Error-resilient representation learning on graphs for label

noise tolerance. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 270–280, 2024.

[9] Yuyin Zhou, Xianhang Li, Fengze Liu, Qingyue Wei, Xuxi Chen, Lequan Yu, Cihang Xie, Matthew P Lungren, and Lei Xing. L2b: Learning to bootstrap robust models for combating label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23523–23533, 2024.

[10] Noo-ri Kim, Jin-Seop Lee, and Jee-Hyong Lee. Learning with structural labels for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27610–27620, 2024.

[11] Boyan Gao, Henry Gouk, and Timothy M Hospedales. Searching for robustness: Loss learning for noisy classification tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6670–6679, 2021.

[12] Lizhi Liao, Heng Li, Weiyi Shang, and Lei Ma. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(3):1–40, 2022.

[13] Kumar Shubham, Aishwarya Jayagopal, Syed Mohammed Danish, Prathosh Ap, and Vaibhav Rajan. Wiser: Weak supervision and supervised representation learning to improve drug response prediction in cancer. *arXiv preprint arXiv:2405.04078*, 2024.

[14] Jintao Meng, Peng Chen, Mohamed Wahib, Mingjun Yang, Liangzhen Zheng, Yanjie Wei, Shengzhong Feng, and Wei Liu. Boosting the predictive performance with aqueous solubility dataset curation. *Scientific Data*, 9(1):71, 2022.

[15] Jina Kim, Jeffrey Willette, Bruno Andreis, and Sung Ju Hwang. Robust molecular property prediction via densifying scarce labeled data. *arXiv preprint arXiv:2506.11877*, 2025.

[16] Esther Heid, Charles J McGill, Florence H Vermeire, and William H Green. Characterizing uncertainty in machine learning for chemistry. *Journal of Chemical Information and Modeling*, 63(13):4012–4029, 2023.

[17] Hunter Lang, Aravindan Vijayaraghavan, and David Sontag. Training subset selection for weak supervision. *Neurips*, 35:16023–16036, 2022.

[18] Fabrice Muhlenbach, Stéphane Lallich, and Djamel A Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, 2004.

[19] Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pages 540–550. PMLR, 2020.

[20] Shengchao Liu, Yanjing Li, Zhuoxinran Li, Zhiling Zheng, Chenru Duan, Zhi-Ming Ma, Omar Yaghi, Animashree Anandkumar, Christian Borgs, Jennifer Chayes, et al. Symmetry-informed geometric representation for molecules, proteins, and crystalline materials. *Advances in neural information processing systems*, 36:66084–66101, 2023.

[21] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[22] Robin Winter, Marco Bertolini, Tuan Le, Frank Noe, and Djork-Arné Clevert. Unsupervised learning of group invariant and equivariant representations. *Advances in Neural Information Processing Systems*, 35:31942–31956, 2022.

[23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[24] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[25] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[26] Arushi Goel, Yunlong Jiao, and Jordan Massiah. Pars: Pseudo-label aware robust sample selection for learning with noisy labels. *arXiv preprint arXiv:2201.10836*, 2022.

[27] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[28] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International conference on machine learning*, pages 7164–7173. PMLR, 2019.

[29] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64 (3):107–115, 2021.

[30] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *Advances in neural information processing systems*, 30, 2017.

[31] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2019.

[32] Devraj Mandal, Shrisha Bharadwaj, and Soma Biswas. A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1381–1390, 2020.

[33] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1657–1667, 2022.

[34] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM international conference on multimedia*, pages 1405–1413, 2021.

[35] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022.

[36] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

[37] Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*, 2022.

[38] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.

[39] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

[40] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

[41] Dario Garcia-Garcia and Robert C Williamson. Divergences and risks for multiclass experiments. In *Conference on Learning Theory*, pages 28–1. JMLR Workshop and Conference Proceedings, 2012.

[42] Prem K Goel and Josep Ginebra. When is one experiment 'always better than'another? *Journal of the Royal Statistical Society Series D: The Statistician*, 52(4):515–537, 2003.

[43] Roy Turgeman and Tom Tirer. Does the data processing inequality reflect practice? on the utility of low-level tasks. *arXiv preprint arXiv:2512.21315*, 2025.

[44] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480, 2007.

[45] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

[46] Yifei Wang, Jizhe Zhang, and Yisen Wang. Do generated data always help contrastive learning? *arXiv preprint arXiv:2403.12448*, 2024.

[47] Artem Moskalev, Anna Sepliarskaia, Erik J Bekkers, and Arnold WM Smeulders. On genuine invariance learning without weight-tying. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 218–227. PMLR, 2023.

[48] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.

[49] Aarti Singh, Clayton Scott, and Robert Nowak. Adaptive hausdorff estimation of density level sets. 2009.

[50] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. *Advances in Neural Information Processing Systems*, 27, 2014.

[51] Henry Reeve and Ata Kabán. Fast rates for a knn classifier robust to unknown asymmetric label noise. In *International Conference on Machine Learning*, pages 5401–5409. PMLR, 2019.

[52] Volume of an n-ball. `https://en.wikipedia.org/wiki/Volume_of_an_n-ball`, 2025. Accessed: 2025-10-07.

[53] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[54] Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pages 540–550. PMLR, 2020. Supplementary Material, Section B.3.

[55] Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. Learning from binary labels with instance-dependent noise. *Machine Learning*, 107(8):1561–1595, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions: analyzing the dependence of cutstats on k-NN accuracy, showing the limitation of k-NN in high-dimensional regimes, and demonstrating that invariant representations based on orthogonal and permutation symmetries can improve subset selection under label noise. These claims are supported by the theoretical results and the empirical evaluations on synthetic datasets, Rotated MNIST, Tetris and CIFAR-10N.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: we discusses limitations in the conclusion and in the invariant representation discussion. In particular, the proposed approach depends on how accurately the underlying group invariance is captured and on whether the learned invariant representation preserves class-discriminative information.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We clearly state the assumptions required for our theoretical results, including the noisy-label setting, balanced class-prior assumptions used for the simplified cutstats analysis, and the invariance assumptions on the true labeling function. Our theoretical results are presented as numbered lemmas, propositions, and corollaries, and the corresponding proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the necessary information to reproduce our main experimental results. This includes the synthetic data-generating functions, invariant functions, noise-generation process, subset-selection ratio, baselines, downstream architectures, datasets, and evaluation metrics. We also provide additional experimental and training details in the appendix, along with an anonymous code repository.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access to our code. Our experiments use synthetic datasets generated from the functions described in the paper and standard benchmark datasets such as Rotated MNIST and CIFAR-10N. We include sufficient experimental details and instructions to support faithful reproduction of the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the main training and evaluation details needed to understand our results, including the datasets, invariant functions, generating functions, noise model, subset-selection ratio, baselines, downstream classifiers, and evaluation metrics. Additional implementation and training details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report results with error bars for the main experimental comparisons. In particular, the tables report mean performance across three different random seeds, along with the corresponding variability, for both classifier accuracy and subset accuracy.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided necessary details in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and believe that our research conforms to it. Our work focuses on theoretical analysis and empirical evaluation of subset selection under label noise using synthetic and standard benchmark datasets, and does not violate neurips code of ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper discusses the positive motivation for improving data quality, particularly for reliable downstream learning and scientific analysis. To the best of our knowledge our work does not have and negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not release models or datasets. The released artifact is code for reproducing our subset-selection experiments, and the datasets used are either synthetically generated or standard benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: Our work does not release any models or datasets,. The released artifact is code for reproducing our subset-selection experiments, and the datasets used are either synthetically generated or standard benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not conduct new crowdsourcing experiments or human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve new research with human subjects, participant recruitment, or new crowdsourced data collection. Therefore, IRB approval or an equivalent review process is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of our core methodology. Our work focuses on k-NN, cutstats, invariant representations, and subset selection under label noise. LLM is used only for formatting purpose.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix Table of Contents

Contents

A Notations **22**

B Training Procedure **22**

C Algorithm **23**

D Definitions **23**
D.1 Group and Associated Details 23
D.2 k-NN bound 24
D.3 Bayes optimal classifier and K-NN 24

E Assumptions **24**
E.1 Performance of a Downstream Classifer [17] 24
E.2 Bayes Optimal Classifier and k-NN [19] 25

F Theoretical Analysis **26**
F.1 Performance of a k-NN 26
F.2 Bayes Optimal Classifier and k-NN 27
F.2.1 Lemma 27
F.2.2 Analysis for bayes classifier 29
F.3 Impossibility Condition and Group Analysis 29
F.4 Discrete group analysis 30

G Experiment on SE(3) Group **30**

H Analysis of SE(3) and other Lie Group **31**

I Comparison with Robustness-Based Baseline **31**

J Performance on Asymmetric Noises **32**

K Performance of K-NN for High Dimension d **33**

L Subset Accuracy across Baselines **33**

M Ablation Study on Subset Size **34**

N Ablation Study on Size of Nearest Neighbor **35**

A Notations

Table 6: Summary of key notations used throughout the paper.

Notation	Description
General setting	
$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$	Noiseless dataset with $x_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$ drawn from $\mathcal{P}$.
$\mathcal{D}_{\text{noisy}} = \{(x_i, \hat{y}_i)\}$	Noisy-labeled dataset; $\hat{y}_i$ denotes the corrupted label.
$\mathcal{S} \subseteq \mathcal{D}_{\text{noisy}}$	Subset selected by *cutstats* for downstream training.
$\boldsymbol{f} : \mathbb{R}^d \to \{0, 1\}$	True labeling function, with $y = \boldsymbol{f}(x)$.
m, d	Training dataset size and feature dimension.
Groups and invariant representations	
$G,\ g \cdot x,\ \mathcal{O}_x$	Symmetry group, its action on x, and the orbit $\mathcal{O}_x = \{g \cdot x : g \in G\}$.
$\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^m$	Invariant representation satisfying $\mathcal{F}(x) = \mathcal{F}(x')$ for $x' \in \mathcal{O}_x$.
$\hat{\mathcal{F}}$	Learned or approximate invariant representation.
$SO(d),\ S_d$	Special orthogonal and symmetric groups, respectively.
***Cutstats* and k-NN**	
$\text{NN}(x),\ k$	Set of k nearest neighbors of x and the number of neighbors used.
$w_{i,j}$	Similarity weight between samples x_i and x_j.
J_iS	Weighted neighbor-label disagreement.
μ_i, σ_i^2, z_i	Null mean, variance, and *cutstats* score $z_i = (J_i - \mu_i)/\sigma_i$.
τ	Retained fraction of examples after ranking by z_i.
y^{knn}	Majority label predicted by k-NN.
Noise and performance	
α, γ	Error rates for class 0 and class 1, respectively, as defined in Lemma 3.1.
$\alpha_{\text{noisy}}, \gamma_{\text{noisy}}$	Noise rates on the complete noisy dataset.
$\alpha_{\mathcal{S}}, \gamma_{\mathcal{S}}$	Noise rates within the selected subset $\mathcal{S}$.
λ_0, λ_1	Class-wise accuracies of k-NN.
$\text{err}_{\text{bal}}(\mathcal{M}, y)$	Balanced error of classifier $\mathcal{M}$ with respect to true labels y.
$\phi,\ \mathcal{M},\ \mathcal{M}^*,\ \Gamma$	Abstention label, downstream classifier, noiseless optimal classifier, and hypothesis class.
k-NN–Bayes analysis	
$\eta_k(x), \eta^*(x)$	k-NN and Bayes-optimal decision rules.
R_X, R^*	Risks of the k-NN and Bayes classifiers.
$n, \mathcal{C}$	Number of correctly labeled samples and set of corrupted samples in Lemma 3.4.
ν, β, ρ	Confidence, Tsybakov low-noise, and Hölder smoothness parameters.
$K_l(d), K_u(d)$	Lower- and upper-bound constants for the admissible range of k in Lemma 3.4.
$K, K', \Delta(\mathcal{C})$	Constants in the risk bounds and minimum clean–corrupted feature separation.
λ	Bound term $\sqrt{\frac{\log n + \log(1/\nu)}{k}} + \left(\frac{k}{n}\right)^{\rho/d}$.

B Training Procedure

Synthetic Dataset Generation: For each group, the dataset is generated according to the data generation functions specified in Table 1 of the main draft. Each sample (x_i) is drawn from a uniform distribution, and its class label (y) is determined by thresholding the output of the data-generating function $h(x)$ using $\mathbf{1}\big(h(x) \geq \theta\big)$. The feature dimension d is set to hundred for Orthogonal groups and five for permutation group. For the Orthogonal group, $h(x)$ is thresholded at 0, while for the Permutation group, the threshold is set based on the mean score. For the Orthogonal group, we set $(c_1, c_2, c_3, k_1, k_2, k_3) = (1, 2, 3, 2, 3, 4)$. For the Permutation group, the hyperparameter l is set to 5. For *cutstats*, we set $\tau = 40\%$ and used a standard k value of 20 for all experiments, following Lang et al. [17]. The prior probability for the synthetic experiments is set to $\mathbb{P}(\hat{y} = 1) = \mathbb{P}(\hat{y} = 0) = \frac{1}{2}$.

Experiments on Rotated MNIST and Tetris: The experiments on the Rotated MNIST and Tetris datasets are conducted using the standard codebase provided by Winter et al. [22], with an 80–20% split between the training and test sets. The unsupervised invariant representation is trained for 20 epochs using the Adam optimizer with a learning rate of 1×10^{-3}. The same hyperparameters are used consistently for both the group-invariant and contrastive representations. The architecture proposed by Winter et al. [22] is adopted for representation learning in both variants—ours (inv) and ours (contrastive). In cases where the original architecture was unavailable, we constructed an equivalent model with the same number of layers and activation functions. For both datasets, the prior probabilities are set according to the class distributions described in Lang et al. [17]. All experiments were conducted on two NVIDIA A6000 GPUs with 48GB of VRAM each.

Downstream Model Training: For the synthetic dataset experiments, we use a two-layer neural network with a hidden dimension of 32 units and ReLU activation. The final output layer employs a sigmoid activation function. All models are trained using three different random seeds, a learning rate of 10^{-2}, and for 20 epochs with a batch size of 1024, optimized using binary cross-entropy loss. For the Rotated MNIST dataset, we use a ResNet-9 classifier for downstream training. For the Tetris dataset, we employ a graph neural network (GNN) with three graph convolutional layers for classification.

Noise: The noisy labels ($\hat{y}$) are generated by randomly reassigning each sample's true label to one of the remaining classes with a predefined noise probability, as specified in the main draft. Experiments on the synthetic dataset are conducted with a noise probability of 0.45 to simulate highly noisy conditions.

C Algorithm

Algorithm 1 Proposed Method

Require: Noisy Dataset ($\mathcal{D}_{\text{train}}$), Group ($G$), class label ($\hat{y}$), Percentage of Training Dataset to consider for the downstream model(τ).

Define an invariant function $\mathcal{F}$ for the group G such that:

$$\mathcal{F}(x_i) = \mathcal{F}(x_k), \quad \forall x_k \in g \cdot x_i \mid g \in G, \quad \text{where } x \in \mathcal{D}_{\text{train}}.$$

for $i \in [1, \ldots, |\mathcal{D}_{\text{train}}|]$ **do**
  Calculate the feature representation $\mathcal{F}(x_i)$ for sample x_i.
end for
for $i \in [1, \ldots, |\mathcal{D}\text{train}|]$ **do**
  for $j \in [1, \ldots, |\mathcal{D}\text{train}|]$ **do**
    Compute the weight $w_{i,j}$ between samples x_i and x_j.
  end for
end for
for $\hat{y}_i \in \hat{y}$ **do**
  Calculate the class probability $P(\hat{y}_i)$.
end for
for $i \in [1, \ldots, |\mathcal{D}_{\text{train}}|]$ **do**
  Compute the z_i score for sample x_i
end for
Identify the top $\tau\%$ samples with the smallest z_i scores and use them for downstream model.

D Definitions

D.1 Group and Associated Details

Definition D.1 (Group)**.** A *group* is a set G with a binary operation $\cdot$ that satisfies:

- **Closure**: $a, b \in G \Rightarrow a \cdot b \in G$,
- **Associativity**: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$,

- **Identity**: $\exists e \in G$ such that $e \cdot a = a \cdot e = a$ for all $a \in G$,
- **Inverse**: $\forall a \in G, \exists a^{-1} \in G$ such that $a \cdot a^{-1} = e$.

Definition D.2 (Group Action)**.** A *group action* of a group G on a set X is a function $\cdot : G \times X \to X$ satisfying:

- **Identity**: $e \cdot x = x$ for all $x \in X$, where e is the identity of G.
- **Compatibility**: $(gh) \cdot x = g \cdot (h \cdot x)$ for all $g, h \in G$ and $x \in X$.

Definition D.3 (Orbit)**.** Let G be a group acting on a set X. The *orbit* of an element $x \in X$ under the action of G is the set:

$$\mathcal{O}_x = \{g \cdot x \mid g \in G\}.$$

It consists of all elements in X that can be reached from x by applying elements of G.

Definition D.4 (Invariant Representation Function)**.** Let G be a group acting on a set X, and let $\mathcal{O}_x$ denote the orbit of $x \in X$ under this action (see Definition D.3). An *invariant representation function* is a mapping

$$\mathcal{F} : X \to \mathbb{R}^d$$

such that

$$\mathcal{F}(x) = \mathcal{F}(x') \quad \text{whenever } x' \in \mathcal{O}_x.$$

That is, $\mathcal{F}$ is constant on each orbit.

D.2 k-NN bound

Definition D.5 (Balanced Error)**.** [17] The balanced error of a classifier f on labels y is defined as:

$$\begin{aligned} \text{err}_{\text{bal}}(f, y) = \frac{1}{2}\big(&\mathbb{P}[f(x) = 0 \mid y = 1] \\ &+ \mathbb{P}[f(x) = 1 \mid y = 0]\big). \end{aligned} \tag{3}$$

Definition D.6 (Error Probabilities)**.** The probability of error in class labels in the dataset w.r.t to the true class label $y \in \{0, 1\}$ is defined as follows:

$$\alpha = \mathbb{P}[y = 0 \mid \hat{y} = 1], \quad \gamma = \mathbb{P}[y = 1 \mid \hat{y} = 0] \tag{4}$$

Definition D.7 (Accuracy of K-NN)**.** The accuracy of K-NN ($\mathcal{H} : x \to y^{knn}$) used for subset selection for true class label $y \in \{0, 1\}$ is defined as follows:

$$\lambda_0 = \mathbb{P}[y^{knn} = 0 \mid y = 0], \quad \lambda_1 = \mathbb{P}[y^{knn} = 1 \mid y = 1] \tag{5}$$

D.3 Bayes optimal classifier and K-NN

Definition D.8 (Minimum pairwise distance)**.** For corrupted samples $\mathcal{C}$, the minimal distance between features of any two samples is defined as follows:

$$\Delta(\mathcal{C}) := \min_{x, x' \in \mathcal{C}, x \neq x'} |x - x'|.$$

E Assumptions

E.1 Performance of a Downstream Classifer [17]

In this section, we list all the assumptions made by Lang et al. [17] to derive the bound relating the performance of a classifier trained on a noiseless dataset to that of the same classifier when trained on a subset of a noisy dataset.

We assume that the nearest neighbors of size k are considered in *cutstats*, with a dataset having equal noise priors $\mathbb{P}(\hat{y} = 0) = \frac{1}{2}$ for a binary classification problem. The selection of a given sample is determined by the label shared among its neighbors.

Assumption E.1 (Nearest Neighbor)**.** Let $\mathcal{D} = \{(x_i, \hat{y}_i)\}_{i=1}^n$ denote a dataset used in *cutstats*, where $\hat{y}_i \in \{0, 1\}$ represents the noisy class label and y_i the true class labels. We assume that class priors are balanced, i.e.,

$$\mathbb{P}(\hat{y} = 0) = \mathbb{P}(\hat{y} = 1) = \tfrac{1}{2}.$$

$$\mathbb{P}(y = 0) = \mathbb{P}(y = 1) = \tfrac{1}{2}.$$

For each sample x_i, *cutstats* considers its k-nearest neighbors $\mathcal{N}_k(x_i)$ according to a predefined distance metric. The selection mechanism assigns smaller z-values to samples whose majority of neighbors share the same label, i.e.,

$$z_i \downarrow \text{ as } \frac{1}{k} \sum_{x_j \in \mathcal{N}_k(x_i)} \mathbb{I}[\hat{y}_j = \hat{y}_i] \uparrow .$$

Our analysis therefore, focuses on the statistical behavior of these k-nearest neighbors and their corresponding labels under this balanced prior setting.

Next, we assume that the predictions made by the underlying heuristic and the noisy labels are conditionally independent given the true class label of an input sample [17, 48].

Assumption E.2 (Conditional Independence)**.** The original work of Lang et al. [17] assumes that the features used to train the model and those used to predict noisy labels ($\hat{y}$) are conditionally independent given the true class label (y). Inspired by this, we assume that the prediction made by the k-NN classifier and the noisy class label assigned to a given example are conditionally independent given the (unobserved) true label y. Intuitively, this assumption ensures that the features used to generate noisy labels (e.g., via crowdsourcing platforms or other heuristic methods) are distinct from the features used for subset selection. Further, this condition is expected to hold whenever the input features used to learn the invariant representation remain uncorrupted by the underlying label noise, which can cover many practical scenarios. However, violation of such a condition by adversarial noise or instance-dependent noise can result in high Bayes error and can impact optimal subset selection. Formally,

$$\mathbb{P}(\hat{y}, y^{\text{knn}} \mid y) = \mathbb{P}(\hat{y} \mid y) \times \mathbb{P}(y^{\text{knn}} \mid y). \tag{6}$$

E.2 Bayes Optimal Classifier and k-NN [19]

We next describe some basic assumptions adopted in previous works [19, 49] that form the foundation for our analysis of the k-NN and Bayes optimal classifiers.

Assumption E.3 (Support Regularity)**.** There exist constants $\omega > 0$ and $r_0 > 0$ such that

$$\text{Vol}(\mathcal{X} \cap B(x, r)) \geq \omega \cdot \text{Vol}(B(x, r))$$

for all $x \in \mathcal{X}$ and $0 < r < r_0$, where $B(x, r) := \{x' \in \mathcal{X} : \|x - x'\| \leq r\}$.

This assumption ensures that the input feature space $\mathcal{X}$ associated with the feature distribution $\mathcal{P}$ does not become arbitrarily thin in any region, thereby guaranteeing sufficient local density for neighborhood-based methods such as k-NN.

If p_X is the probability density function associated with ($\mathcal{P}$), then the next assumption ensures that for sufficiently large samples, there is a good coverage of the feature space. [19]

Assumption E.4. (p_X bounded from below).

$$p_{X,0} := \inf_{x \in \mathcal{X}} p_X(x) > 0.$$

Finally, we assume the smoothness of $\eta(x)$ [19, 50, 51].

Assumption E.5. (η Hölder continuous) *There exists* $0 < \rho \leq 1$ *and* $C_\rho > 0$ *such that for class conditional distribution* $\big(\eta(x) = \mathbb{P}(y = 1 \mid X = x)\big)$

$$|\eta(x) - \eta(x')| \leq C_\alpha |x - x'|^\rho \quad \text{for all } x, x' \in \mathcal{X}.$$

F Theoretical Analysis

F.1 Performance of a k-NN

Proposition 3.2. *[Performance of k-NN] Consider the setup of Lemma 3.1. for k-NN ($\mathcal{H} : x \mapsto y^{knn}$), with y^{knn} as the class label of the majority of the samples in the nearest neighbor and*

$$\lambda_0 = \mathbb{P}(y^{knn} = 0 \mid y = 0), \qquad \lambda_1 = \mathbb{P}(y^{knn} = 1 \mid y = 1)$$

denote the class-wise accuracies of k-NN for class 0 *and class* 1*, respectively. Then the error rate of the subset selected by k-NN satisfies:*

$$\alpha_{\mathcal{S}} = \frac{\alpha_{noisy}(1-\lambda_0)}{\alpha_{noisy}(1-\lambda_0) + (1-\alpha_{noisy})\lambda_1}, \qquad \gamma_{\mathcal{S}} = \frac{\gamma_{noisy}(1-\lambda_1)}{\gamma_{noisy}(1-\lambda_1) + (1-\gamma_{noisy})\lambda_0}$$

Proof. As per *cutstats*, for any subset selected using k-NN, a data point is likely to be included in the subset $\mathcal{S}$ if the label assigned to the sample matches the majority label among its nearest neighbors (y^{knn}), as stated in Assumption E.1.

$$\alpha_{\mathcal{S}} = \mathbb{P}[y = 0 \mid \hat{y}_{\mathcal{S}} = 1], \quad \text{where } \hat{y}_{\mathcal{S}} = 1 \text{ denotes samples in subset } \mathcal{S} \ (\hat{y} = 1, y^{\text{knn}} = 1) \tag{7}$$

$$= \frac{\mathbb{P}(\hat{y} = 1, y^{\text{knn}} = 1 \mid y = 0) \cdot \mathbb{P}(y = 0)}{\sum_{c \in \{0,1\}} \mathbb{P}(\hat{y} = 1, y^{\text{knn}} = 1 \mid y = c) \cdot \mathbb{P}(y = c)} \quad \text{(by Bayes' rule).} \tag{8}$$

Consider:

$$\begin{aligned}
\mathbb{P}(\hat{y} = 1, y^{\text{knn}} = 1 \mid y = 0) &= \mathbb{P}(\hat{y} = 1 \mid y = 0) \cdot \mathbb{P}(y^{\text{knn}} = 1 \mid y = 0) \quad \text{(by Assumption E.2)} \\
&= \frac{\mathbb{P}(y = 0 \mid \hat{y} = 1) \cdot \mathbb{P}(\hat{y} = 1)}{\mathbb{P}(y = 0)} \cdot (1 - \lambda_0) \\
&= \frac{\alpha_{noisy} \cdot \mathbb{P}(\hat{y} = 1) \cdot (1 - \lambda_0)}{\mathbb{P}(y = 0)}, \quad \text{where } \alpha_{noisy} = \mathbb{P}(y = 0 \mid \hat{y} = 1)
\end{aligned} \tag{9}$$

for non-abstained data.

Similarly,

$$\begin{aligned}
\mathbb{P}(\hat{y} = 1, y^{\text{knn}} = 1 \mid y = 1) &= \mathbb{P}(\hat{y} = 1 \mid y = 1) \cdot \mathbb{P}(y^{\text{knn}} = 1 \mid y = 1) \quad \text{(by Assumption E.2)} \\
&= \frac{\mathbb{P}(y = 1 \mid \hat{y} = 1) \cdot \mathbb{P}(\hat{y} = 1)}{\mathbb{P}(y = 1)} \cdot \lambda_1 \\
&= \frac{(1 - \alpha_{noisy}) \cdot \mathbb{P}(\hat{y} = 1) \cdot \lambda_1}{\mathbb{P}(y = 1)}.
\end{aligned} \tag{10}$$

Substituting Equations 9 and 10 into Equation 8, and assuming equal class priors, i.e., $\mathbb{P}(y = 0) = \mathbb{P}(y = 1)$, we obtain:

$$\alpha_{\mathcal{S}} = \frac{\alpha_{noisy}(1-\lambda_0)}{\alpha_{noisy}(1-\lambda_0) + (1-\alpha_{noisy})\lambda_1}.$$

Similarly, for the other class:

$$\gamma_{\mathcal{S}} = \frac{\gamma_{noisy}(1-\lambda_1)}{\gamma_{noisy}(1-\lambda_1) + (1-\gamma_{noisy})\lambda_0}. \tag{11}$$

□

Corollary 3.3. *Suppose the class priors are balanced, i.e.,* $\mathbb{P}(y=0)=\mathbb{P}(y=1)=\frac{1}{2}$. *If the k-NN classifier has error* $\mathbb{P}(y^{knn}\neq y)\leq\frac{1}{2}$, *then the subset error rates satisfy*

$$\alpha_{\mathcal{S}}\leq\alpha_{noisy}\quad \textit{and}\quad \gamma_{\mathcal{S}}\leq\gamma_{noisy}.$$

Moreover, as $\lambda_0,\lambda_1\to 1$, *the subset error rates also reduces:*

$$\alpha_{\mathcal{S}},\,\gamma_{\mathcal{S}}\;\to\;0.$$

Proof. By Proposition 3.2,

$$\alpha_{\mathcal{S}}=\frac{\alpha_{noisy}(1-\lambda_0)}{\alpha_{noisy}(1-\lambda_0)+(1-\alpha_{noisy})\lambda_1},\qquad \gamma_{\mathcal{S}}=\frac{\gamma_{noisy}(1-\lambda_1)}{\gamma_{noisy}(1-\lambda_1)+(1-\gamma_{noisy})\lambda_0}.$$

With balanced priors, the overall k-NN accuracy is

$$\mathbb{P}(y^{\text{knn}}=y)=\tfrac{1}{2}\lambda_0+\tfrac{1}{2}\lambda_1=\tfrac{1}{2}(\lambda_0+\lambda_1).$$

The assumption $\mathbb{P}(y^{\text{knn}}\neq y)\leq\frac{1}{2}$ is equivalent to $\mathbb{P}(y^{\text{knn}}=y)\geq\frac{1}{2}$, hence

$$\lambda_0+\lambda_1\;\geq\;1\quad\Longleftrightarrow\quad 1-\lambda_0\;\leq\;\lambda_1\quad\text{and}\quad 1-\lambda_1\;\leq\;\lambda_0.$$

Consider the function $f(a)=\dfrac{au}{au+(1-a)v}$ for $a\in(0,1)$ and $u,v>0$. A direct algebraic check shows

$$f(a)\leq a\iff au\leq a^2u+a(1-a)v\iff a(1-a)(u-v)\leq 0\iff u\leq v.$$

Apply this with $a=\alpha_{noisy}$, $u=1-\lambda_0$, $v=\lambda_1$. Since $1-\lambda_0\leq\lambda_1$, we get $\alpha_{\mathcal{S}}\leq\alpha_{noisy}$. Likewise, with $a=\gamma_{noisy}$, $u=1-\lambda_1$, $v=\lambda_0$ and $1-\lambda_1\leq\lambda_0$, we get $\gamma_{\mathcal{S}}\leq\gamma_{noisy}$.

For the limit claim, let $\lambda_0,\lambda_1\to 1$. Then $1-\lambda_0\to 0$ and $1-\lambda_1\to 0$, so from the formulas above,

$$\alpha_{\mathcal{S}}\;=\;\frac{\alpha_{noisy}(1-\lambda_0)}{\alpha_{noisy}(1-\lambda_0)+(1-\alpha_{noisy})\lambda_1}\;\longrightarrow\;0,\qquad \gamma_{\mathcal{S}}\;\longrightarrow\;0.$$

□

F.2 Bayes Optimal Classifier and k-NN

In this section, we show how leveraging underlying invariances can enable the k-NN classifier to perform comparably to the Bayes optimal classifier. Before proceeding, we establish two important lemmas that form the foundation of our proof. The first lemma demonstrates that the volume of a unit ball decreases with increasing feature dimension d.

F.2.1 Lemma

Lemma F.1 (Super-exponential decay of the unit ball volume [52, 53])**.** *Let* V_d *denote the volume of the unit-radius Euclidean ball in* $\mathbb{R}^d$*:*

$$V_d=\frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}.$$

where, Γ *denotes gamma function. Then* $V_d\to 0$ *as* $d\to\infty$, *and moreover* V_d *decays super-exponentially, that is,*

$$\forall\, c>1\;\;\exists\, D \textit{ such that } d\geq D\implies V_d\leq c^{-d}.$$

Proof. Using the standard Stirling lower bound for the Gamma function,

$$\Gamma(x+1)\geq\sqrt{2\pi x}\left(\frac{x}{e}\right)^x,\qquad x>0.$$

Setting $x=\frac{d}{2}$, we have

$$\Gamma\left(\frac{d}{2}+1\right)\geq\sqrt{\pi d}\left(\frac{d}{2e}\right)^{\frac{d}{2}}.$$

Substituting this into the formula for V_d gives

$$V_d = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} \leq \frac{\pi^{d/2}}{\sqrt{\pi d}\left(\frac{d}{2e}\right)^{d/2}} = \frac{1}{\sqrt{\pi d}}\left(\frac{2\pi e}{d}\right)^{d/2}.$$

Taking logarithms,

$$\log V_d \leq -\frac{1}{2}\log(\pi d) + \frac{d}{2}\left(\log(2\pi e) - \log d\right) = -\frac{d}{2}\log d + O(d).$$

Therefore,

$$V_d \leq \exp\left(-\tfrac{1}{2}d\log d + O(d)\right).$$

For any fixed $c > 1$, since $\frac{1}{2}\log d \to \infty$, we can find D such that for all $d \geq D$,

$$\tfrac{1}{2}\log d - C \geq \log c$$

for some constant C absorbing the $O(1)$ term. Hence for all $d \geq D$,

$$\log V_d \leq -d\log c,$$

i.e. $V_d \leq c^{-d}$. Thus, V_d decays faster than any exponential c^{-d}, proving the super-exponential rate.

□

The next lemma formally establishes how, for data exhibiting permutation invariance, the sorting function influences the induced density.

Lemma F.2. *Let $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ be an absolutely continuous random vector with density p_X. Assume p_X is* permutation–invariant*, i.e.*

$$p_X(x_1, \ldots, x_d) = p_X\left(x_{\pi(1)}, \ldots, x_{\pi(d)}\right) \quad \textit{for every permutation } \pi \in S_d.$$

Let $\mathrm{sort}(x)$ be the non-decreasing rearrangement of $x = (x_1, \ldots, x_d)$, and set $Y = \mathrm{sort}(X)$. Then for every y with $y_1 < y_2 < \cdots < y_d$,

$$p_Y(y) = d!\, p_X(y).$$

Proof. Let $C = \{y \in \mathbb{R}^d : \ y_1 < \cdots < y_d\}$. For any $y \in C$, the preimage of y under the sorting map consists of the $d!$ permutations,

$$\mathrm{sort}^{-1}(y) = \{\pi(y) : \ \pi \in S_d\}.$$

More generally, for measurable $A \subseteq C$ we have the (essentially disjoint) decomposition

$$\mathrm{sort}^{-1}(A) = \bigsqcup_{\pi \in S_d} \pi(A),$$

Therefore,

$$\Pr(Y \in A) = \Pr\left(X \in \mathrm{sort}^{-1}(A)\right) = \sum_{\pi \in S_d} \Pr\left(X \in \pi(A)\right) = \sum_{\pi \in S_d} \int_{\pi(A)} p_X(x)\, dx.$$

Each permutation $\pi : \mathbb{R}^d \to \mathbb{R}^d$ is a linear isometry with Jacobian determinant ± 1, hence it is measure-preserving. By change of variables,

$$\int_{\pi(A)} p_X(x)\, dx \ = \ \int_A p_X\left(\pi(y)\right) dy.$$

Using permutation invariance, $p_X(\pi(y)) = p_X(y)$ for all $\pi \in S_d$, so

$$\Pr(Y \in A) = \sum_{\pi \in S_d} \int_A p_X(y)\, dy \ = \ d! \int_A p_X(y)\, dy.$$

Since this holds for every measurable $A \subseteq C$, the integrand identifies the density of Y on C:

$$p_Y(y) = d!\, p_X(y) \qquad \text{for a.e. } y \in C.$$

Finally, the boundary $\{y_i = y_j \text{ for some } i < j\}$ has Lebesgue measure zero, so the same equality holds almost everywhere on $\{y_1 \leq \cdots \leq y_d\}$. □

F.2.2 Analysis for bayes classifier

Since our proof builds upon the theoretical framework of Bahri et al. [19], we first restate their main result for completeness. According to their analysis, if a dataset contains n uncorrupted samples and a subset $\mathcal{C}$ of samples with corrupted (noisy) labels, then for a k-nearest neighbors classifier $\eta_k(x)$ and the Bayes-optimal classifier $\eta^*(x)$, the following relationship holds:

Lemma 3.4. *[19] Let $\nu > 0$, and assume the conditions of Bahri et al. [19] hold under the Tsybakov noise condition with parameter β. Then there exist constants $K_l(d), K_u(d), K, K' > 0$, depending only on the dimension d and the underlying distribution, such that with probability at least $1 - \nu$:*

If k lies in the range

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n,$$

Then,

$$\mathbb{P}\big(\eta_k(x) \neq \eta^*(x)\big) \leq K \cdot \lambda^\beta, \qquad R_X - R^* \leq K' \cdot \lambda^{\beta+1},$$

where

$$\lambda = \left(\sqrt{\frac{\log n + \log(1/\nu)}{k}} + \left(\frac{k}{n}\right)^{\rho/d} \right),$$

R_X and R^* denote the risks of the k-NN and the Bayes optimal classifier, respectively.

where $\Delta(\mathcal{C})$ denotes the minimum distance between features of corrupted examples (Definition D.8) and ρ is a parameter associated with Holder's continuity (Assumption E.5).

F.3 Impossibility Condition and Group Analysis

Proposition 3.5. *[High-dimensional k-NN and Bayes Classifier] As the feature dimension d increases, the factor $K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$ in Lemma 3.4 decreases with d $\big(K_u(d)$ decreases faster than any fixed exponential base with $d\,\big)$. Consequently, there exists a threshold d_0 such that for all data dimension $d \geq d_0$ and for a given n, ν, ρ and Δ no k satisfies*

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n.$$

Proof. According to Bahri et al. [54], we have

$$K_u(d) = \omega \cdot v_d \cdot p_{x,0},$$

Where v_d denotes the volume of the unit ball in $\mathbb{R}^d$, $p_{x,0}$ is the lower bound on the data density (as defined in Assumption E.4), and ω is the regularity constant from Assumption E.3.

$K_l(d)$ grows approximately linearly with d [54]. However, the unit ball volume v_d decreases faster than exponentially as d increases (see Lemma F.1). Therefore, $K_u(d)$ decays super-exponentially with d, causing the upper bound

$$K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$$

to shrink rapidly with increasing dimension.

Hence, there exists a threshold d_0 such that for all $d > d_0$,

$$K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n < K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}},$$

implying that no valid k can satisfy Lemma 3.4 the above inequality for the given parameters n, ν, ρ, and Δ. □

Proposition 3.6. *[Orthogonal Group] For data whose true-labeling function satisfy orthogonal invariance ($SO(d)$) as per section 3.1, if k-NN utilizes the invariant representation function ($\mathcal{F}$), and the inequality holds for dimension (d=1), i.e.,*

$$K_l(1) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+1}} \leq k \leq K_u(1) \cdot \Delta(\mathcal{C}) \cdot n.$$

Then there exists a k for all dimensions d for which k-NN will approximate Bayes optimal classifier for a given n, ν, ρ and Δ,

Proof. Under orthogonal invariance ($SO(d)$), the invariant representation function $\mathcal{F}$ can be chosen as the Euclidean norm, i.e., $\mathcal{F}(x) = \|x\|_2$. This mapping reduces any d-dimensional input to a one-dimensional scalar that is invariant to orthogonal transformations.

The difficulty discussed in Proposition 3.5 arises due to the exponential dependence on the ambient dimension d. However, by using the invariant function $\mathcal{F}$, the data are effectively mapped from $\mathbb{R}^d$ to $\mathbb{R}$, eliminating the dimensional dependence responsible for the degradation in $K_u(d)$.

Consequently, if the condition

$$K_l(1) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+1}} \leq k \leq K_u(1) \cdot \Delta(\mathcal{C}) \cdot n$$

is satisfied in the one-dimensional case, it will also hold for data in any dimension d when using the invariant representation $\mathcal{F}(x) = \|x\|_2$. Therefore, the k-NN classifier based on $\mathcal{F}$ will approximate the Bayes optimal classifier for all d. □

Proposition 3.7. *[Permutation Group] For data whose true-labeling function satisfy permutation invariance (S_d) as per section 3.1, if k-NN utilizes the invariant representation function ($\mathcal{F}$), then, $K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$ does not decrease with d, and there will exist k for a given n, ν, ρ and Δ for which*

$$K_l(d) \cdot \log^2(1/\nu) \cdot n^{\frac{\rho}{\rho+d}} \leq k \leq K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n.$$

holds, and k-NN can approximate Bayes optimal classifier.

Proof. Under permutation invariance (S_d), the invariant representation function $\mathcal{F}$ can be taken as the sorting operator, which maps any input vector to a canonical ordered form. As shown in Lemma F.2, this operation can increase the effective probability density by a factor of $d!$, since $d!$ samples in the original data space correspond to a single representation in the invariant (sorted) space.

The constant, $K_u(d) = \omega \cdot v_d \cdot p_{x,0}$ (Proposition 3.5), where v_d is the volume of the d-dimensional unit ball, ω is a regularity constant and $p_{x,0}$ is the lower bound of the data density. While v_d decreases super-exponentially with d, the sorting operation effectively multiplies $p_{x,0}$ by a $d!$ factor, compensating for this decay.

Therefore, the product $K_u(d) \cdot \Delta(\mathcal{C})^d \cdot n$ does not diminish with increasing d. Consequently, there exists a valid k satisfying the inequality in Lemma 3.4, and the k-NN classifier based on the permutation-invariant representation $\mathcal{F}$ can approximate the Bayes optimal classifier. □

F.4 Discrete group analysis

Although Lemma F.2 is established for absolutely continuous distributions, a similar analysis can be extended to discrete distributions under the assumption that the coordinates of each sample are distinct. In the discrete setting, the admissible range of k is given by $d \log n \lesssim k \lesssim n p_{\min}$, where $p_{\min}$ denotes the minimum probability mass. For a support of size q^d, this range is non-empty only when $n \gtrsim q^d d \log n$. Under permutation invariance, each sorted configuration represents an orbit containing $d!$ distinct elements, increasing the effective minimum probability mass by a factor of $d!$. Consequently, the sample-size requirement reduces to $n \gtrsim \frac{q^d d \log n}{d!}$. Thus, analogous to Proposition 3.7, exploiting permutation symmetry preserves the effective dimension while increasing the effective probability mass, resulting in a reduction in the required sample size by a factor of $d!$. However, when coordinates are repeated, the orbit size is smaller than $d!$, and the corresponding improvement is reduced.

G Experiment on SE(3) Group

Table 7 presents the results for the Tetris dataset that satisfy SE(3) invariance. The proposed method ("Ours (group-inv)") demonstrates robustness to noise even on datasets with other symmetries, achieving classification accuracies of 75.72%, 67.28%, and 40.95% across the three noise levels, respectively. These results highlight its superior capability in handling label noise in settings where other subset selection methods largely fail.

Table 7: Performance comparison on Tetris dataset under different noise probabilities.

Method	Noise 0.4		Noise 0.6		Noise 0.8	
	Classifier-Acc	Subset-Acc	Classifier-Acc	Subset-Acc	Classifier-Acc	Subset-Acc
NoiseLess	88.30 ± 0.33	100.00 ± 0.00	88.30 ± 0.33	100.00 ± 0.00	88.30 ± 0.33	100.00 ± 0.00
Noisy-Complete	$\mathbf{88.18 \pm 0.33}$	59.95 ± 0.64	$\mathbf{75.23 \pm 1.55}$	40.09 ± 0.59	34.48 ± 5.49	20.21 ± 0.24
Random	54.47 ± 4.14	60.21 ± 1.22	44.93 ± 1.00	40.56 ± 0.40	24.18 ± 5.55	20.33 ± 0.59
Vanilla Cutstats	57.05 ± 3.77	60.69 ± 1.04	50.88 ± 7.66	40.48 ± 0.72	29.62 ± 6.43	20.72 ± 0.31
Entropy	33.80 ± 5.96	59.56 ± 0.85	37.35 ± 0.33	40.32 ± 1.18	28.63 ± 5.19	19.95 ± 0.49
Forget	24.53 ± 5.72	38.05 ± 6.07	20.92 ± 6.25	26.21 ± 7.58	12.40 ± 0.71	15.00 ± 2.03
Herding	33.08 ± 5.38	59.73 ± 1.61	32.90 ± 5.63	40.58 ± 0.52	32.95 ± 7.07	20.22 ± 0.56
Ours (contrastive)	52.12 ± 1.86	60.23 ± 0.53	50.90 ± 2.52	39.52 ± 0.85	30.18 ± 4.46	20.05 ± 0.63
Ours (group inv)	75.72 ± 5.65	$\mathbf{100.00 \pm 0.00}$	67.28 ± 6.33	$\mathbf{95.11 \pm 1.45}$	$\mathbf{40.95 \pm 5.34}$	$\mathbf{36.17 \pm 0.84}$

H Analysis of SE(3) and other Lie Group

Our theoretical analysis establishes two mechanisms through which symmetry-induced invariance can improve the k-NN approximation to the Bayes-optimal classifier: reducing the effective dimensionality of the representation space and increasing its effective data density, as characterized by Propositions 3.6 and 3.7, respectively. These observations provide a general criterion for the utility of invariance across different symmetry groups. For finite groups, invariance increases the effective density in proportion to the orbit size, whereas compact positive-dimensional Lie groups can yield both dimensionality reduction and an additional orbit-volume factor. In contrast, non-compact groups generally contribute through dimensionality reduction without a finite global orbit-volume factor. In the Tetris setting, the relevant symmetries are described by $SE(3) \times S_c$ acting on c points. Specifically, centering eliminates three translational degrees of freedom, rotational invariance removes up to three additional dimensions and introduces a rotational orbit-volume factor, while permutation invariance can increase the effective density by a factor of $c!$. Thus,

$$\underbrace{\omega\, v_{3c}\, p_{x,0}}_{\text{raw features}} < \underbrace{\omega\, v_{3c-3}\, p_{x,0}}_{+\text{ translations}} < \underbrace{\omega\, v_{3c-6}\, \rho\, p_{x,0}}_{+\text{ rotations}} < \underbrace{\omega\, v_{3c-6}\, c!\, \rho\, p_{x,0}}_{+\text{ permutations}}$$

Consequently, the theoretical framework extends beyond $SE(3)$ to other Lie groups, with the resulting approximation gains determined by the reduction in effective dimension and the increase in effective density induced by the corresponding invariant representation. Importantly, these benefits depend on the extent to which the learned representation successfully captures the underlying symmetries. Rigorous analysis of these and other groups is left for future work.

I Comparison with Robustness-Based Baseline

We have primarily compared our method with subset-selection based baselines, as they are most closely aligned with our approach and are designed to improve the quality of the training dataset. However, for benchmarking purposes , we additionally include a comparison with the Generalized Cross Entropy (GCE) loss [24], a model-robustness-based method. Since GCE does not modify or improve the training data itself, we report only classifier accuracy for this baseline. As shown in Tables 8–10, our method continues to outperform the robustness-based loss function in most settings across all three benchmarks.

These results highlight the distinct objectives and applicability of symmetry-aware subset selection and robust training methods. Our approach is particularly advantageous in high-dimensional settings with known underlying symmetries, especially when the downstream classifier is unknown at the time of data curation or when obtaining a reliable subset is itself the primary objective. Unlike GCE, which improves a specific model's robustness to label noise, our method explicitly identifies and removes potentially unreliable samples, producing a curated dataset that can be reused across different downstream tasks. However, the benefits of symmetry-aware selection depend on the validity of the assumed symmetries and may be less pronounced in low-dimensional settings or when a single downstream model can be trained robustly. This is reflected in the competitive performance of GCE on the Tetris benchmark (Table 10), suggesting that the two approaches offer complementary advantages depending on the data characteristics and intended application.

Table 8: Classifier accuracy on Orthogonal Group and Permutation Group tasks.

Method	Orthogonal Group	Permutation Group
Complete noisy + GCE	56.43 ± 2.23	81.17 ± 0.024
Ours	$\mathbf{68.52 \pm 0.66}$	$\mathbf{86.00 \pm 0.84}$

Table 9: Classifier accuracy across noise levels on Rot-MNIST.

Method	Noise 0.4	Noise 0.6	Noise 0.8
Complete noisy + GCE	87.38 ± 1.96	85.23 ± 1.03	61.57 ± 0.64
Ours (contrastive)	83.95 ± 2.04	76.28 ± 0.57	41.52 ± 3.71
Ours (group inv)	$\mathbf{91.81 \pm 0.58}$	$\mathbf{89.37 \pm 1.83}$	$\mathbf{75.73 \pm 5.99}$

Table 10: Classifier accuracy across noise levels on Tetris.

Method	Noise 0.4	Noise 0.6	Noise 0.8
Complete noisy + GCE	$\mathbf{88.18 \pm 0.33}$	$\mathbf{75.65 \pm 1.75}$	38.38 ± 0.63
Ours (contrastive)	52.12 ± 1.86	50.90 ± 2.52	30.18 ± 4.46
Ours (group inv)	75.72 ± 5.65	67.28 ± 6.33	$\mathbf{40.95 \pm 5.34}$

J Performance on Asymmetric Noises

Tables 11–13 present the results under asymmetric label noise. For the synthetic datasets, we use noise rates of 0.4 for class 0 and 0.2 for class 1, following the experimental setup in Menon et al. [55]. For Rotated MNIST and Tetris, we use the protocol in Han et al. [27] to construct pairwise asymmetric noise with noise probability 0.45.

Across these settings, Ours (group inv) consistently outperforms the baselines in subset accuracy and achieves stronger classifier accuracy, demonstrating that knowledge of the underlying symmetry helps recover high-quality subsets and train better downstream classifiers.

Table 11: Performance under asymmetric noise for the orthogonal and permutation groups, with noise rates $[0.4, 0.2]$ for classes 0 and 1, respectively. Results are reported as mean $\pm$ standard deviation across three seeds.

Method	Orthogonal Group		Permutation Group	
	Classifier-Acc	Subset-Acc	Classifier-Acc	Subset-Acc
NoiseLess	84.93 ± 5.17	100.00 ± 0.00	94.58 ± 2.03	100.00 ± 0.00
Noisy-Complete	68.82 ± 0.40	73.92 ± 0.12	81.97 ± 3.43	69.45 ± 0.05
Vanilla Cutstats	45.78 ± 2.91	74.73 ± 0.95	$\mathbf{90.48 \pm 0.29}$	98.89 ± 0.08
Random	68.43 ± 0.54	73.71 ± 0.43	80.22 ± 6.38	69.26 ± 0.42
Entropy	68.45 ± 0.53	77.31 ± 0.74	58.80 ± 0.72	78.23 ± 0.50
Forget	68.53 ± 0.72	74.16 ± 0.50	86.95 ± 3.32	79.79 ± 1.51
Herding	67.27 ± 1.24	72.28 ± 0.86	67.92 ± 1.48	71.62 ± 0.38
Ours	$\mathbf{73.92 \pm 1.83}$	$\mathbf{99.85 \pm 0.07}$	90.38 ± 0.72	$\mathbf{99.64 \pm 0.06}$

Table 12: Performance under asymmetric noise for Rotated MNIST.

Method	Classifier Acc. (%)	Subset Acc. (%)
Noiseless Complete	97.45 ± 0.15	100.00 ± 0.00
Noisy Complete	59.87 ± 1.04	55.01 ± 0.00
Vanilla CutStats	59.28 ± 2.72	68.47 ± 0.39
Random	53.23 ± 2.74	55.12 ± 0.39
Entropy	57.54 ± 4.68	59.05 ± 0.15
Forget	55.72 ± 2.91	59.94 ± 0.26
Herding	54.01 ± 3.57	55.23 ± 0.24
Ours (contrastive)	55.14 ± 5.20	57.24 ± 0.18
Ours (group inv)	$\mathbf{71.87 \pm 1.45}$	$\mathbf{72.19 \pm 0.39}$

Table 13: Performance under asymmetric noise for Tetris.

Method	Classifier Acc. (%)	Subset Acc. (%)
Noiseless Complete	88.20 ± 0.33	100.00 ± 0.00
Noisy Complete	57.00 ± 6.40	55.05 ± 0.01
Vanilla CutStats	58.20 ± 6.58	55.12 ± 0.41
Random	53.98 ± 11.24	54.91 ± 0.16
Entropy	33.28 ± 4.87	55.21 ± 0.16
Forget	32.37 ± 5.85	56.47 ± 2.33
Herding	44.73 ± 5.22	55.16 ± 0.33
Ours (contrastive)	$\mathbf{63.60 \pm 9.28}$	54.72 ± 0.87
Ours (group-inv)	62.32 ± 8.12	$\mathbf{74.80 \pm 1.22}$

K Performance of K-NN for High Dimension d

We conducted an additional ablation comparing our method with **vanilla Cutstats** across different data dimensions for both the orthogonal and permutation groups. Tables 14 and 15 report the accuracy of K-NN in recovering the clean subset when using the original features (vanilla Cutstats), and our invariant features. The results clearly show that as the dimensionality increases, the performance of K-NN with vanilla Cutstats degrades significantly, whereas using invariant representations helps maintain strong recovery performance.

Table 14: Accuracy of k-NN across data dimensions d for the orthogonal group.

Method	$d = 200$	$d = 400$	$d = 600$	$d = 800$	$d = 1000$	$d = 2048$	$d = 4096$
Vanilla Cutstats	73.59 ± 2.59	70.73 ± 1.26	65.96 ± 3.83	62.90 ± 3.30	55.35 ± 0.33	55.19 ± 1.78	60.37 ± 0.75
Ours	$\mathbf{73.97 \pm 2.52}$	$\mathbf{76.47 \pm 0.99}$	$\mathbf{75.65 \pm 1.03}$	$\mathbf{73.30 \pm 2.26}$	$\mathbf{75.31 \pm 1.83}$	$\mathbf{73.82 \pm 2.14}$	$\mathbf{77.03 \pm 0.68}$

Table 15: Accuracy of k-NN across data dimensions d for the permutation group.

Method	$d = 200$	$d = 400$	$d = 600$	$d = 800$	$d = 1000$	$d = 2048$	$d = 4096$
Vanilla Cutstats	57.08 ± 0.53	56.57 ± 0.73	56.15 ± 0.50	55.84 ± 1.04	55.91 ± 0.77	56.17 ± 0.54	56.45 ± 0.77
Ours	$\mathbf{70.03 \pm 1.19}$	$\mathbf{72.52 \pm 0.90}$	$\mathbf{71.27 \pm 1.64}$	$\mathbf{72.21 \pm 2.11}$	$\mathbf{72.07 \pm 1.66}$	$\mathbf{70.67 \pm 3.02}$	$\mathbf{75.13 \pm 1.00}$

L Subset Accuracy across Baselines

Table 16 presents a comparison of our method with existing baselines, particularly subset-selection methods, across different subset sizes for the orthogonal and permutation groups. The results demonstrate that our method consistently outperforms the baselines across the evaluated subset sizes.

Table 16: Comparison of subset accuracy across different subset sizes for orthogonal and permutation groups. Results are reported for different subset selection methods, with our method highlighted in **bold**.

Method	Orthogonal Group				Permutation Group			
	20	**40**	**60**	**80**	**20**	**40**	**60**	**80**
NoiseLess	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Noisy-Complete	55.08	55.08	55.08	55.08	55.33	55.33	55.33	55.33
Random	63.97	54.93	64.59	64.65	62.29	55.50	62.39	62.25
Vanilla CutStats	65.30	60.51	65.79	65.22	88.25	71.21	80.90	73.14
Entropy	67.15	56.46	65.54	65.17	68.89	56.05	65.12	63.74
Forget	65.64	58.69	65.89	65.76	72.01	63.04	68.40	64.97
Herding	64.72	55.66	64.35	64.43	63.92	54.13	62.89	62.88
Ours	**87.77**	**74.31**	**82.42**	**75.94**	**90.15**	**74.42**	**83.41**	**74.44**

M Ablation Study on Subset Size

Tables 17,18, 19 provide K-NN ablations for Rotated MNIST, as well as the orthogonal and permutation group on settings described in the main draft. For each noise level, we vary the value of τ, the subset size for a fixed value of k=20, to study its effect on both subset accuracy and classifier accuracy.

Table 17: Ablation study: Performance comparison across different subset sizes for K=20 under varying noise probabilities for rotated MNIST for classifier accuracy (Cls-Acc) and subset accuracy (Sub-Acc). The table reports the mean across three different seeds.

Noise Prob	Method	20%		40%		60%		80%	
		Cls-Acc	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**
0.6	**Ours (group inv)**	**87.10**	**99.34**	**89.37**	**90.51**	**89.73**	**64.83**	**80.47**	**49.57**
	Ours (contrastive)	58.59	64.82	76.28	53.30	82.62	47.79	77.37	43.73
0.8	**Ours (group inv)**	**63.58**	**53.04**	**75.73**	**37.77**	**69.37**	**29.53**	**69.76**	**24.03**
	Ours (contrastive)	35.51	27.06	41.52	24.24	50.24	22.39	52.49	21.21

Table 18: Ablation study on the orthogonal group: Performance comparison across different subset sizes under varying noise probabilities. The table reports the mean across three different seeds.

Noise Prob	20%		40%		60%		80%	
	Cls-Acc	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**
0.30	65.53	99.94	72.80	99.91	75.35	99.59	73.78	87.48
0.45	60.72	80.56	68.48	74.31	68.58	68.26	67.10	62.47

Table 19: Ablation study on the permutation group: Performance comparison across different subset sizes under varying noise probabilities. The table reports the mean across three different seeds.

Noise Prob	Method	20%		40%		60%		80%	
		Cls-Acc	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**	**Cls-Acc**	**Sub-Acc**
0.30	Ours	89.12	99.96	90.77	99.76	91.90	98.88	93.78	87.31
0.45	Ours	86.85	80.60	88.32	74.42	82.93	68.42	80.63	62.65

N Ablation Study on Size of Nearest Neighbor

Tables 20 and 21 provide K-NN ablations for Rotated MNIST, as well as the orthogonal and permutation group on settings described in the main draft. For each noise level, we vary the value of K to study its effect on both subset accuracy and classifier accuracy.

Lemma 3.4 establishes that the admissible range of k depends on the noise level and the intrinsic properties of the data distribution. In sufficiently high-dimensional spaces, this range may become empty, whereas an appropriate invariant representation can reduce the effective dimensionality and restore a non-empty admissible range. This theoretical insight explains the relatively low sensitivity of our method to the choice of k. In practice, k should also be sufficiently large to ensure that the neighborhood of a mislabeled sample contains a diverse set of examples, enabling reliable detection of label inconsistencies.

Table 20: Ablation study: Performance comparison across different K-NN values under varying noise probabilities for classifier accuracy (Cls-Acc) and subset accuracy (Sub-Acc). The table reports the mean across three different seeds.

Noise Prob	Method	K=20		K=40		K=80		K=160		K=320	
		Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc
0.6	**Ours (group inv)**	**89.37**	**90.51**	**91.23**	**93.40**	**90.34**	**94.29**	**90.82**	**94.43**	**87.40**	**93.91**
	Ours (contrastive)	76.28	53.30	68.70	56.29	65.49	58.90	70.19	60.86	62.65	62.14
0.8	**Ours (group inv)**	**75.73**	**37.77**	**76.21**	**42.57**	**80.95**	**45.92**	**76.33**	**47.44**	**76.30**	**48.07**
	Ours (contrastive)	41.52	24.24	43.23	25.47	49.79	26.85	52.00	28.49	46.13	30.15

Table 21: Ablation study: Performance comparison across different K-NN values for orthogonal and permutation group settings. The table reports classifier accuracy (Cls-Acc) and subset accuracy (Sub-Acc) across three different seeds for 0.45 noise probability

Group	Method	K=20		K=40		K=80		K=160		K=320	
		Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc	Cls-Acc	Sub-Acc
Orthogonal group	**Ours**	68.52	71.40	69.04	77.64	68.88	83.80	69.20	87.82	71.28	89.81
Permutation group	**Ours**	84.68	71.52	87.94	76.36	87.25	81.14	88.78	85.25	88.28	87.96