

829 A Proof of Theorems

830 A.1 Proof of Theorem 4.1

831 According to the condition on $\mathcal{E}_Y = \mathcal{E}_y$ and $\mathcal{Q}_Y = \mathcal{Q}_y$, i.e., $\mathcal{U}(Y) = \mathcal{U}(y)$ and $\mathcal{V}(Y) = \mathcal{V}(y)$,
 832 where

$$\mathcal{V}(Y) = \frac{PY}{\|PY\|} \in \mathbb{R}^{(n+1)d}, \quad \mathcal{U}(Y) = (I_{(n+1)d} - P)Y \in \mathbb{R}^{(n+1)d}.$$

833 we have

$$\begin{aligned} & \mathcal{U}(Y) = \mathcal{U}(y) \\ \Leftrightarrow & (I_{(n+1)d} - P)Y = \mathcal{U}(y) \\ \Leftrightarrow & Y = \mathcal{U}(y) + \mathcal{V}(Y)z \\ \Leftrightarrow & Y = \mathcal{U}(y) + \mathcal{V}(y)z \quad (\because \mathcal{V}(Y) = \mathcal{V}(y)) \\ \Leftrightarrow & Y = a + bz, \end{aligned}$$

834 where $a = \mathcal{U}(y)$, $b = \mathcal{V}(y)$, and $z = T(Y) = \|\eta_y^T Y\|_2^2 = \|PY\|_2^2$.

835 Then, we have

$$\begin{aligned} & \{Y \in \mathbb{R}^{(1+n)d} \mid \mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(y)\} \\ = & \{Y \in \mathbb{R}^{(1+n)d} \mid \mathcal{E}_Y = \mathcal{E}_y, Y = a + bz, z \in \mathbb{R}\} \\ = & \{Y = a + bz \in \mathbb{R}^{(1+n)d} \mid \mathcal{E}_{a+bz} = \mathcal{E}_y, z \in \mathbb{R}\} \\ = & \{Y = a + bz \in \mathbb{R}^{(1+n)d} \mid z \in \mathcal{Z}\}, \end{aligned}$$

836 where \mathcal{Z} is the truncation region defined as

$$\mathcal{Z} = \{z \in \mathbb{R} \mid \mathcal{E}_{a+bz} = \mathcal{E}_y\}.$$

837 Therefore, by noting that $\|\eta_y^T s\|_2^2$ is zero, we obtain

$$T(Y) \mid \{\mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(y)\} \sim \text{TC}(\text{tr}(P), \mathcal{Z}),$$

838 where $\text{TC}(\text{tr}(P), \mathcal{Z})$ is a truncated χ^2 -distribution with the degrees of freedom $(1+n)d$, whose
 839 domain is the truncation region \mathcal{Z} .

840 A.2 Proof of Theorem 4.2

841 The sampling distribution of the test statistic conditional on $\mathcal{E}_Y = \mathcal{E}_y$ and $\mathcal{Q}(Y) = \mathcal{Q}(y)$ denoted by

$$T(Y) \mid \{\mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(y)\}$$

842 is a truncated χ^2 -distribution with the degrees of freedom $(1+n)d$ and the truncation region \mathcal{Z} defined
 843 in Theorem 4.1. Thus, by applying the probability integral transform, under the null hypothesis,

$$p_{\text{selective}} \mid \{\mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(y)\} \sim \text{Unif}(0, 1),$$

844 which leads to

$$\mathbb{P}_{H_0}(p_{\text{selective}} \leq \alpha \mid \mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(y)) = \alpha, \quad \forall \alpha \in (0, 1).$$

845 Next, for any $\alpha \in (0, 1)$, we have

$$\begin{aligned} & \mathbb{P}_{H_0}(p_{\text{selective}} \leq \alpha \mid \mathcal{E}_Y = \mathcal{E}_y) \\ &= \int \mathbb{P}_{H_0}(p_{\text{selective}} \leq \alpha \mid \mathcal{E}_Y = \mathcal{E}_y, \mathcal{Q}(Y) = \mathcal{Q}(Y)) \mathbb{P}_{H_0}(\mathcal{Q}(Y) = \mathcal{Q}(Y) \mid \mathcal{E}_Y = \mathcal{E}_y) d\mathcal{Q}(y) \\ &= \alpha \int \mathbb{P}_{H_0}(\mathcal{Q}(Y) = \mathcal{Q}(y) \mid \mathcal{E}_Y = \mathcal{E}_y) d\mathcal{Q}(y) \\ &= \alpha. \end{aligned}$$

846 Therefore, we obtain the result in Theorem 4.2 as follows:

$$\begin{aligned} \mathbb{P}_{H_0}(p_{\text{selective}} \leq \alpha) &= \sum_{\mathcal{E}_y} \mathbb{P}_{H_0}(p_{\text{selective}} \leq \alpha \mid \mathcal{E}_Y = \mathcal{E}_y) \mathbb{P}_{H_0}(\mathcal{E}_Y = \mathcal{E}_y) \\ &= \alpha \sum_{\mathcal{E}_y} \mathbb{P}_{H_0}(\mathcal{E}_Y = \mathcal{E}_y) \\ &= \alpha. \end{aligned}$$

B Selection Event Characterization

In this section, we characterize the selection events $\mathcal{E}_{\mathbf{Y}} = \mathcal{E}_{\mathbf{y}}$ of deep k NN-based anomaly detection (AD). The selection event of deep k NN-based AD consists of two components: the selection event related to the k NN-based AD, and the selection event related to the deep learning models that perform the transformation into latent features. The former is described in Appendix B.1, and the latter in Appendix B.2. Finally, in Appendix B.3, we describe how to identify the data space that satisfies the selection event and how to compute the selective p -values.

B.1 Selection Event for k NN Anomaly Detection

In the selection events of k NN-AD, it is necessary to consider events such as selecting the k nearest instances, the anomaly score exceeding a threshold, and determining k based on the data. In the following, we describe these events one by one. It is worth noting that all the events described below can be collectively represented by a set of linear inequalities, which facilitates the computation of truncation regions for the truncated normal distribution used in selective p -value calculations.

Selection event for k^{th} nearest neighbor The test statistic in Eq. (3) depends on the selection of k^{th} nearest neighbor instance of the test instance \mathbf{X}^{test} . Therefore, the condition on the k^{th} nearest neighbor instance is required. Specifically, by conditioning on

$$\text{dist}(\mathbf{X}^{\text{test}}, \mathbf{X}_{o(k)}) \geq \text{dist}(\mathbf{X}^{\text{test}}, \mathbf{X}_{o(k')}) \quad (7)$$

for $k' = 1, \dots, k-1$, and

$$\text{dist}(\mathbf{X}^{\text{test}}, \mathbf{X}_{o(k)}) \leq \text{dist}(\mathbf{X}^{\text{test}}, \mathbf{X}_{o(k')}) \quad (8)$$

for $k' = k+1, \dots, n$, we can consider only cases where the k -th nearest neighbor is the same as the observed case. Hereafter, the conditions in Eq.(7) and Eq.(8) are collectively represented as $\mathcal{N}_{\mathbf{Y}} = \mathcal{N}_{\mathbf{y}}$.

Selection event for anomaly score Since the statistical test is performed only on test instances selected in the AD, it is essential to consider the selection events associated with it. A test instance is selected and if its anomaly score, as defined in Eq. (1), exceeds a threshold θ . The condition for the anomaly score is written as

$$\log \text{dist}(\mathbf{X}^{\text{test}}, \mathbf{X}_{o(k)}) - \frac{\log k}{d} \geq \theta. \quad (9)$$

With the conditions in Eq.(9), we can characterize the selection event that the test case \mathbf{X}^{test} is selected in AD. Hereafter, the condition in Eq.(9) is represented as $\mathcal{K}_{\mathbf{Y}} = \mathcal{K}_{\mathbf{y}}$.

Selection event for data-driven selection of k In the case of the data-driven option for determining the number of neighbors k , its effect must also be appropriately considered as a selection event. For example, consider the scenario where k_1, \dots, k_K are candidate values for k , and the candidate that maximizes the anomaly score in Eq. (1) is selected. Let the selected $k \in \{k_1, \dots, k_K\}$ be denoted as k^* . Then, the selection event is simply given by $\log \text{dist}(\mathbf{x}^{\text{test}}, \mathbf{x}_{o(k^*)}) - \frac{\log k^*}{d} \geq \log \text{dist}(\mathbf{x}^{\text{test}}, \mathbf{x}_{o(k_t)}) - \frac{\log k_t}{d}, \forall t \in [K]$. In the case of data-driven option to determine k , in addition to the four selection events mentioned above, this event must also be incorporated as an additional condition. Hereafter, we denote this selection event as $\mathcal{S}_{\mathbf{Y}} = \mathcal{S}_{\mathbf{y}}$.

B.2 Selection Event for Deep Learning Models

When using k -nearest neighbors AD with feature representations from a pre-trained deep learning model, the influence of the model should be considered as a selection event. SI for deep learning has been discussed in prior studies, and tools like the software facilitate the analysis of selection events in these models. In this study, we employ methods from earlier research to calculate selective p -values, taking into account selection events related to deep learning models. The basic idea in these methods involves decomposing the model into components and representing each as a piecewise linear function. For example, operations in a CNN such as convolution, ReLU activation, max

pooling, and up-sampling are represented as piecewise linear functions. In the experiment, we utilize the feature representation of a CNN model pre-trained on the ImageNet database. This model is represented precisely as a composition of piecewise linear functions. We explain the selection events regarding the deep learning model that transforms an image instance $\mathbf{x}_i \in \mathbb{R}^d$ to a latent feature vector $\mathbf{z}_i \in \mathbb{R}^{\tilde{d}}$. We consider a deep learning model that consists of sequential piecewise-linear functions (e.g., convolution, ReLU activation, max pooling, and up-sampling). Obviously, the composite function of those piecewise-linear functions maintains its piecewise-linear nature. Thus, within a specific real space in \mathbb{R}^d , the deep learning model simplifies to a linear function, which can be expressed as:

$$\phi_{\text{DL}}(\mathbf{x}_i) = \mathbf{B} + \mathbf{W}\mathbf{x}_i \quad \text{if } \mathbf{x}_i \in \mathcal{P},$$

where $\mathbf{B} \in \mathbb{R}^{\tilde{d}}$ and $\mathbf{W} \in \mathbb{R}^{\tilde{d} \times d}$ represent the bias and weight matrices, and $\mathcal{P} \subseteq \mathbb{R}^d$ is a polytope where ϕ_{DL} acts as a linear function. The polytope can be characterized by a set of linear inequalities. For details on computing these linear inequalities, see [57]. Let us denote the set of polytopes for all instances in \mathbf{Y} as:

$$\mathcal{D}_{\mathbf{Y}} := \{\mathcal{P} \mid \mathbf{X}_i \in \mathbf{Y}, \mathbf{X}_i \in \mathcal{P}\}.$$

Hereafter, we denote the selection event as $\mathcal{D}_{\mathbf{Y}} = \mathcal{D}_{\mathbf{y}}$.

B.3 Computing Selective p -values

Based on the discussions in Appendix B.1 and B.2, selective p -values in (6) can be rewritten as follows:

$$p_{\text{selective}} := \mathbb{P}_{\text{H}_0}(T(\mathbf{Y}) \geq T(\mathbf{y}) \mid \mathcal{N}_{\mathbf{Y}} = \mathcal{N}_{\mathbf{y}}, \mathcal{K}_{\mathbf{Y}} = \mathcal{K}_{\mathbf{y}}, \mathcal{S}_{\mathbf{Y}} = \mathcal{S}_{\mathbf{y}}, \mathcal{D}_{\mathbf{Y}} = \mathcal{D}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{Y}} = \mathcal{Q}_{\mathbf{y}}).$$

Calculating this selective p -values is complex, but we effectively use methods from existing SI research. We specifically use the parametric programming (pp)-based method from previous studies [38]. In SI, statistical inference is based on the probability measure within the subspace \mathcal{Z} of the data space $\mathbb{R}^{(1+n)d}$ where selection event conditions are met. By conditioning on the selection event for the nuisance component, $\mathcal{Q}_{\mathbf{Y}} = \mathcal{Q}_{\mathbf{y}}$, \mathcal{Z} reduces to a one-dimensional subspace (see Theorem 4.1 and its proof in Appendix A.1). The selection events are formulated as unions of intersections of linear or quadratic inequalities, suitable when using L_1 or L_2 distances for k -nearest neighbors. \mathcal{Z} consists of finite number of intervals along a line in the $(1+n)d$ -dimensional space, and the pp-based method systematically enumerates all intervals that meet these conditions.

Since the noise is Gaussian, the test statistic $T(\mathbf{Y})$ under the null hypothesis H_0 follows a one-dimensional truncated Gaussian distribution within the subspace \mathcal{Z} , comprising finite intervals along a line. The selective p -value is calculated as the tail probability of this truncated distribution. Early SI research often simplified calculations by assuming \mathcal{Z} as a single interval under additional conditions, which still controls the false detection probability but reduces detection power. In our problem, a similar simplification can be considered by enforcing \mathcal{Z} to be a single interval. In the experiments in §5, we conduct an ablation study comparing this simple approach (denoted as w/o-pp) as one of the baselines.

C Details of the Experiments

C.1 Additional Type I Error Rate Results

We also conducted experiments to investigate the Type I error rate when the data dimension n , d and the number of neighbors k were varied in the parametric and semi-parametric setting. Specifically, we varied $d \in \{5, 10, 15, 20\}$, $k \in \{1, 2, 5, 10\}$ and $n \in \{100, 200, 500, 1000\}$, while setting the default parameters as $d = 5$, $k = 3$ and $n = 100$. In all cases, we generated the datasets in the same way as in the experiments on synthetic datasets (§5.2). The results are shown in Figures 5, 6 and 7.

To further assess the robustness of our method, we conducted experiments on datasets that deviate from the normal distribution. Specifically, data are sampled from the exponentially modified Gaussian (EMG), generalized normal distribution (GND), skew normal distribution (SND), and Student's t -distribution. The degree of deviation from the normal distribution is quantified using the Wasserstein distance l , and we evaluate the Type I error rate for each case by varying $l \in \{0.01, 0.02, 0.03, 0.04\}$. The results are shown in Figure 8.

936 C.2 Additional Power Results

937 We also conducted experiments to investigate the power when the number of training data n , the data
 938 dimension d and the number of neighbors k are varied in the parametric and semi-parametric setting.
 939 We varied $n \in \{100, 200, 500, 1000\}$, $d \in \{5, 10, 15, 20\}$ and $k \in \{1, 2, 5, 10\}$ while setting the
 940 default parameters as $n = 100$, $d = 5$, $k = 3$ and signal strength $\delta = 5$. Furthermore, we conducted
 941 additional experiments where n and d was varied, considering the case where k was adaptively
 942 selected from $\in \{1, 2, 5, 10\}$ in a data-driven manner. In all cases, we generated the datasets in the
 943 same way as in the experiments on synthetic datasets (§5.2). The results are shown in Figures 9, 10,
 944 11, and 12.

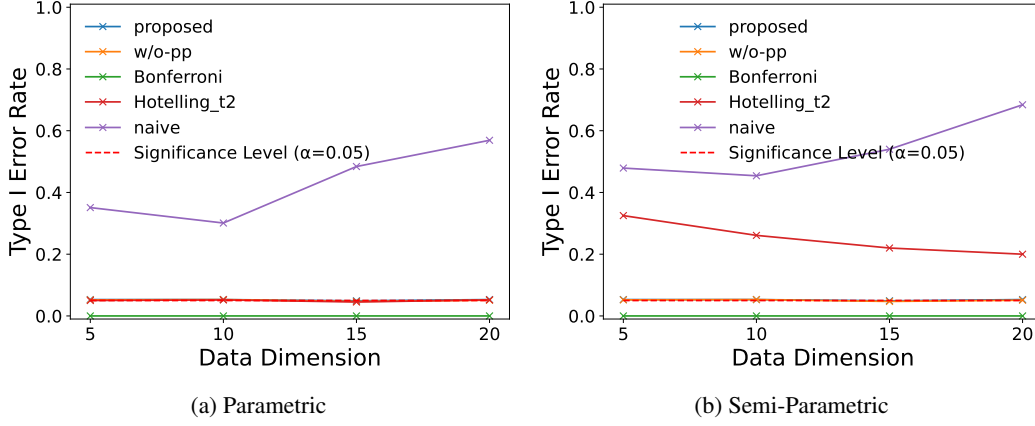


Figure 5: Results of Type I error rate when varying the data dimension d . proposed, w/o-pp, and Bonferroni successfully control the Type I error rate across all settings. naive fails and the results of Bonferroni are almost zero, because it is too conservative. Since Hotelling_t2 does not involve a parameter k , its value remains unchanged. Hotelling_t2 also fails in the semi-parametric setting.

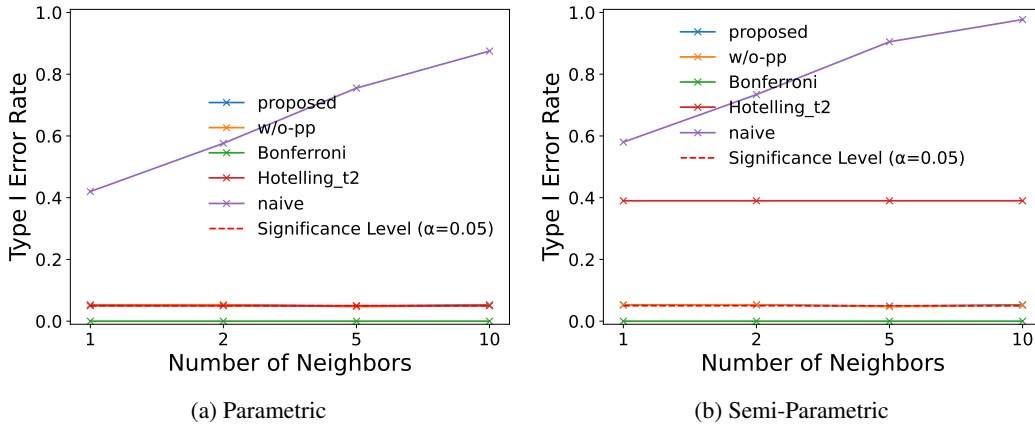


Figure 6: Results of Type I error rate when varying the number of neighbors k . proposed, w/o-pp, and Bonferroni successfully control the Type I error rate across all settings. naive fails and the results of Bonferroni are almost zero, because it is too conservative. Since Hotelling_t2 does not involve a parameter k , its value remains unchanged in the both settings. In the semi-parametric setting, Hotelling_t2 fails to control the Type I error rate.

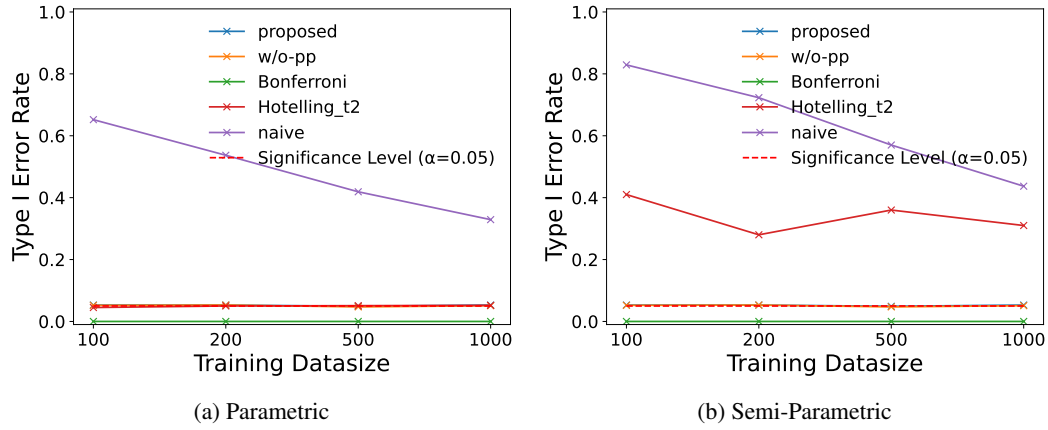


Figure 7: Results of Type I error rate when varying the number of dataset size n . proposed, w/o-pp, and Bonferroni successfully control the Type I error rate across all settings. naive fails and the results of Bonferroni are almost zero, because it is too conservative. Hotelling_t2 also fails in the semi-parametric setting.

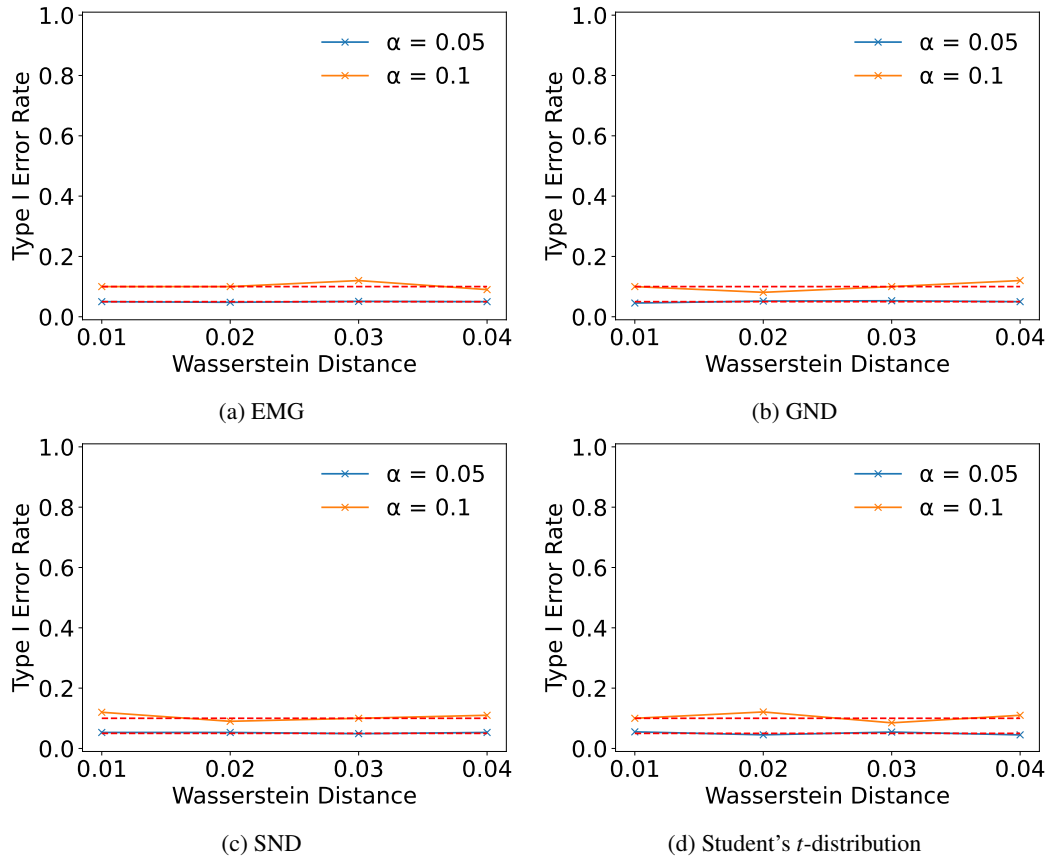


Figure 8: Results of Type I error rate when varying the Wasserstein distance l . proposed successfully control the Type I error rate in both significance levels $\alpha \in \{0.05, 0.1\}$.

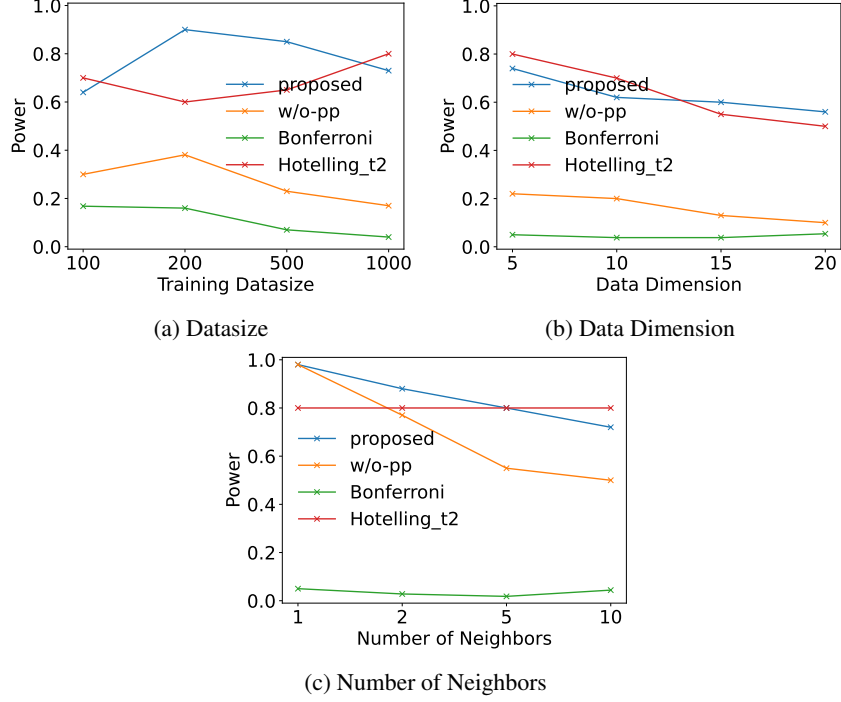


Figure 9: These are in the parametric setting. Power for a fixed number of neighbors k . The results show the effect of varying the training dataset size n , the data dimension d , and k . Our proposed method (proposed) and Hotelling_t2 outperformed other methods across all settings.

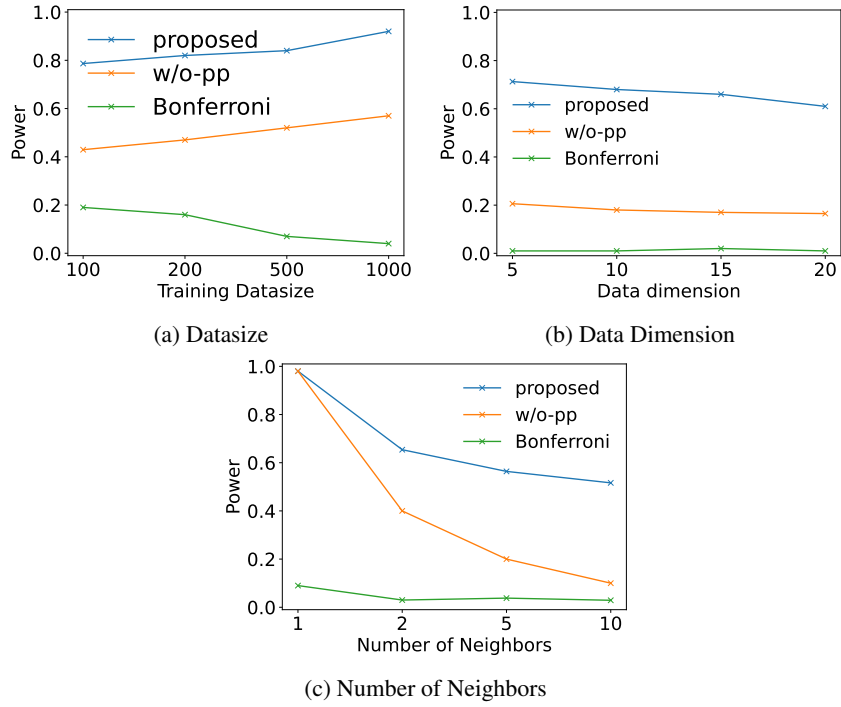


Figure 10: These are in the semi-parametric setting. Power for a fixed number of neighbors k . The results show the effect of varying the training dataset size n , the data dimension d , and k . Our proposed method (proposed) outperformed other methods across all settings.

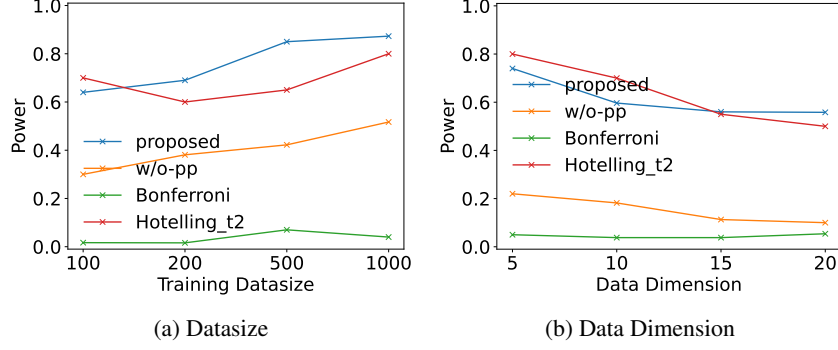


Figure 11: These are in the parametric setting. Power for an adaptively selected number of neighbors k . The results show the effect of varying the training dataset size n and the data dimension d . Our proposed method (proposed) and Hotelling_t2 outperformed other methods across all settings.

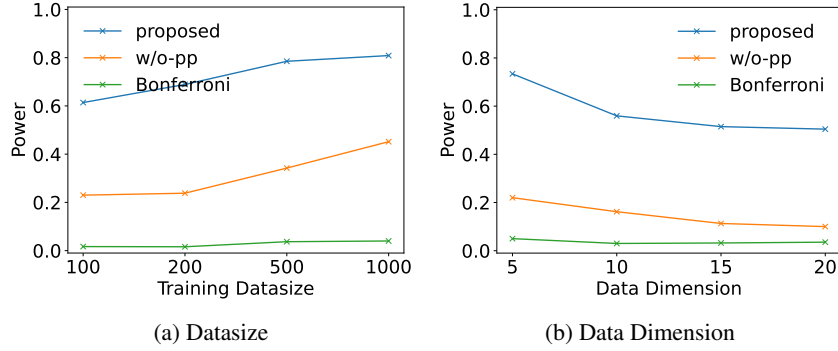


Figure 12: These are in the semi-parametric setting. Power for an adaptively selected number of neighbors k . The results show the effect of varying the training dataset size n and the data dimension d . Our proposed method (proposed) outperformed other methods across all settings.

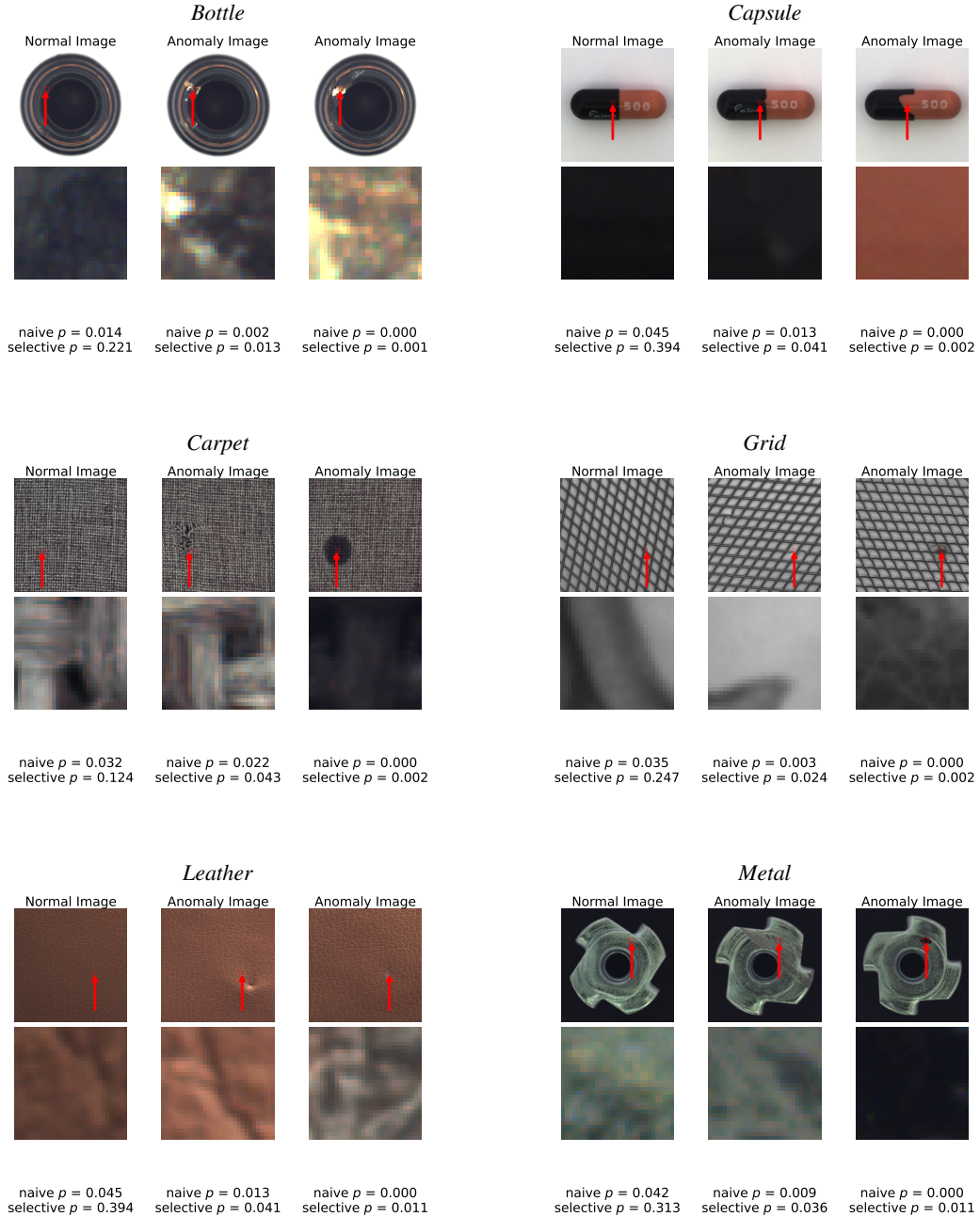
945 C.3 Details of Tabular Datasets

946 We used the following 10 real datasets from the Kaggle Repository. All datasets are licensed under
 947 the CC BY 4.0 license.

- 948 • *Heart*: Dataset for predicting heart attacks
- 949 • *Money*: Dataset on financial transactions in a virtual environment
- 950 • *Fire*: Dataset on fires in the MUGLA region in June
- 951 • *Cancer*: Dataset related to breast cancer diagnosis
- 952 • *Credit*: Dataset on credit card transactions
- 953 • *Student*: Dataset related to student performance
- 954 • *Bankruptcy*: Dataset on company bankruptcies
- 955 • *Drink*: Dataset on the quality of drinking water
- 956 • *Nuclear*: Dataset on pressurized nuclear reactors
- 957 • *Network*: Dataset on anomaly detection in virtual network environments

958 C.4 Experimental Results on Image Data Examples

959 We evaluated proposed and naive on the 10 datasets from MVTec AD dataset. The datasets used
 960 in this study are *Carpet*, *Grid*, *Leather*, *Tile*, *Wood*, *Bottle*, *Capsule*, *Metal Nut*, *Transistor*, and
 961 *Zipper*. Examples from each dataset are shown in Figure 14. In each example, we present patches
 962 corresponding to true negative and true positive cases, along with both the naive p -value and the
 963 selective p -value.



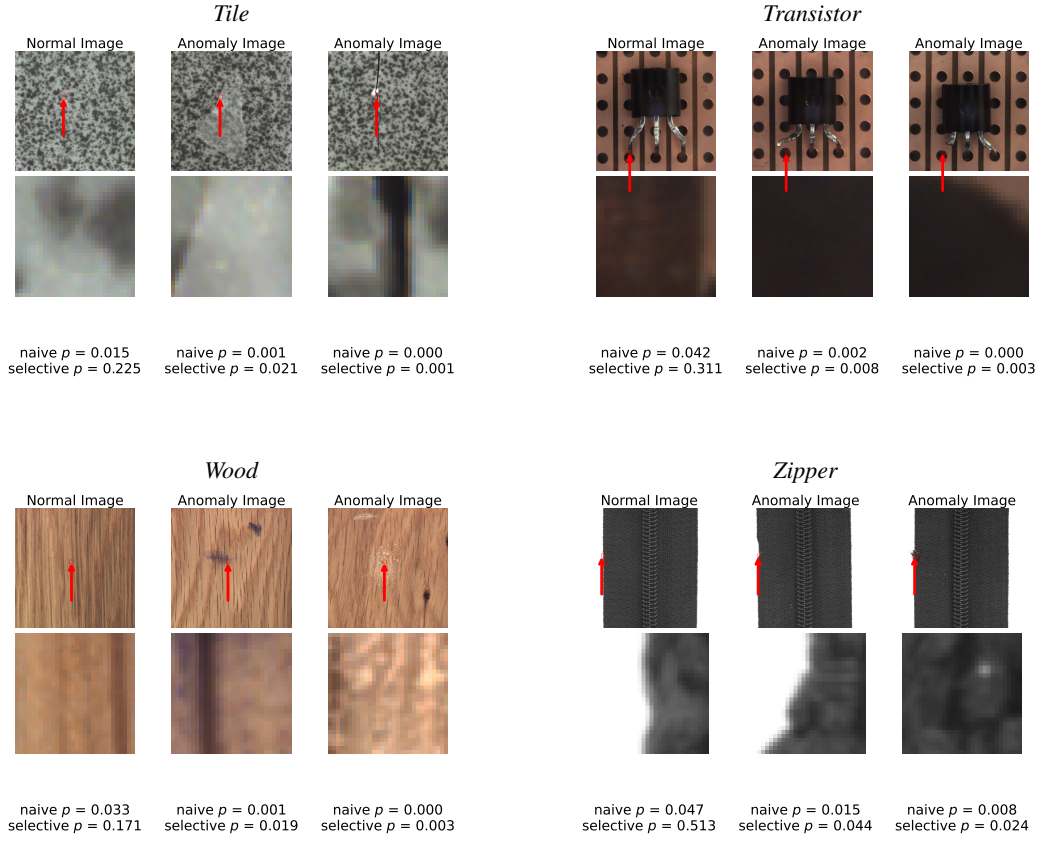


Figure 14: Experimental results of 10 datasets from MVTec AD dataset. For each dataset, one normal example (left) and two anomaly examples (center, right) are showed. For each example, the top row displays the original image used for testing along with the patch location (marked in red), while the bottom row presents the extracted patch image. For all normal examples, the naive p -value is below the significance level $\alpha = 0.05$ (false positive), whereas the proposed selective p -value correctly results in a true negative. For all anomaly examples, the selective p -value successfully detects anomalies.