DeepGD3: Unknown-Aware Deep Generative/Discriminative Hybrid Defect Detector for PCB Soldering Inspection

Abstract

We present a novel approach for detecting soldering defects in Printed Circuit Boards (PCBs) composed mainly of Surface Mount Technology (SMT) components, using advanced computer vision and deep learning techniques. The main challenge addressed is the detection of soldering defects in new components for which only samples of good soldering are available at the model training phase. To address this, we design a system composed of generative and discriminative models to leverage the knowledge gained from the soldering samples of old components to detect the soldering defects of new components. To meet industrial quality standards, we keep the leakage rate (i.e., miss detection rate) low by making the system "unknown-aware" with a low unknown rate. We evaluated the method on a real-world dataset from an electronics company. It significantly reduces the leakage rate from 1.827% $\pm$ 3.063% and 1.942% $\pm$ 1.337% to 0.063% $\pm$ 0.075% with an unknown rate of 3.706% $\pm$ 2.270% compared to the discriminative and generative approaches, respectively.

1 INTRODUCTION

Deep learning has made significant advancements in academia and industries thanks to the abundance of data and the enhancement of computational power. Industries such as manufacturing, medicine, and transportation can cut costs using neural network predictions. Deep learning-based image classification techniques for identifying defects in printed circuit boards are becoming increasingly prevalent in the electronics manufacturing sector. This success is generally through implementing advanced machine vision imaging systems and acquiring abundant training examples that closely resemble the testing examples. However, this requirement can limit the use of deep learning in real-world situations where the testing examples may be new, novel, and dissimilar to the training examples.

Imagine a scenario where the assembly line includes both old and new components. We train the deep learning model on available examples, including the good and defective soldering samples of old components and only good soldering samples of new components. It is then applied directly to detect defective soldering in new components. This approach is limited as it needs to consider that the defective soldering of new components may be dissimilar to the training samples, which can negatively impact the model's performance. Hence, we should consider advanced techniques such as transfer learning, domain adaptation, and meta-learning to improve performance and adapt the model to detect defective soldering in new components. Additionally, to meet the manufacturing standard in real-world industrial applications, it is reasonable to make the model unknown-aware and balance the accuracy and unknown rate. The unknown cases can then be further examined at the next station of the assembly line.

We aim to achieve knowledge transfer and unknown awareness in these situations simultaneously. In [Raina et al., 2003, Fujino et al., 2005, Bosch et al., 2008, Ouyang et al., 2011, Kuleshov and Ermon, 2017, Roth et al., 2018, Grcić et al., 2022, Loh et al., 2022, Cao and Zhang, 2022], two kinds of models, the discriminative and generative models, were combined. Raina et al. [2003] mainly addresses text categorization tasks. It describes a hybrid model in which a high-dimensional subset of the parameters is trained to maximize the generative likelihood, and another subset of parameters is discriminatively trained to maximize the conditional likelihood. Instead, we seek to use deep neural networks to combine discriminative and generative models for our goals. The combined model exchanges knowledge between these two distinct models, forming a shared embedding z, and combines the predictions of these two models to make final predictions.

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

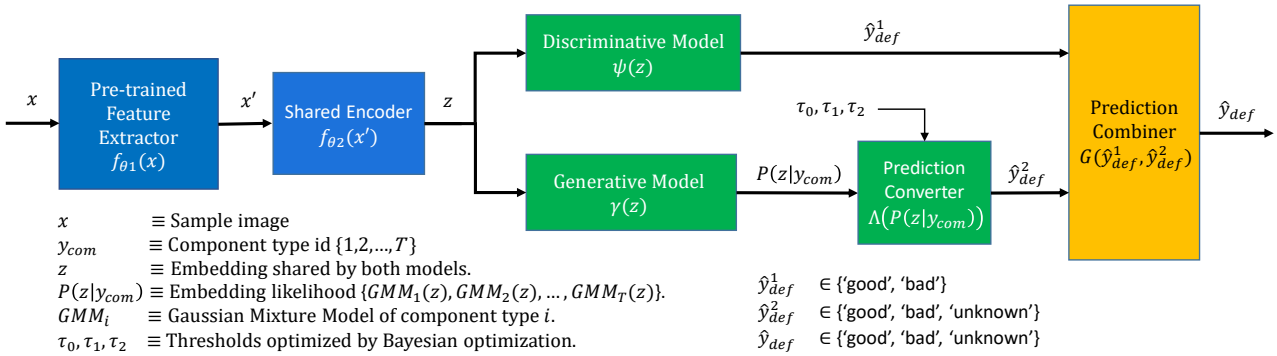


Figure 1: DeepGD3: The unknown-aware **deep G**enerative/**D**iscriminative hybrid **D**efect **D**etector.

We propose a deep neural network with two branches sharing a common feature extractor as depicted in Figure 1. The upper branch is the discriminative defect detector telling the input sample x is good or defective $\hat{y}^1_{def}$; the other is the generative defect detector. The latter produces the likelihood of being a specific component type $P(z|y_{com})$, which is then converter to the second defectiveness prediction $\hat{y}^2_{def}$. A prediction combiner combines these two predictions for the final predictions – "good," "bad," or "unknown."

The shared feature extractor $f_{\theta 2}(x')$ exchanges the knowledge learned by the discriminative defect detector and the knowledge learned by the generative defect detector, making the latter being to classify the soldering as good, defective, or unknown, regardless of the defective soldering of new components not seen at the training stage. The predictions of these two branches are combined to make the final prediction, which is more reliable than only using one branch.

The task we addressed here can be seen as a sub-task of zero-shot learning [Xian et al., 2018]. It is similar to compositional zero-shot learning (CZSL) [Mancini et al., 2022] but not the same. The final prediction of our task is the soldering status only, not the composition of soldering status and component types. This setting comes from the fact that we know the component types in advance in real-world applications. An algorithm targeting this setting should perform better than those targeting the setting of CZSL. Our method considers these facts and considerations, converting and combining two predictions into one prediction, resulting in superior performance. Unknown awareness also makes our setting more practical and different from CZSL.

According to experiments, the proposed method solves the task mentioned above much better than using only the discriminative or generative models. We summarize our contributions as follows:

- The introduction of a new task for the electronic assembly line, where the goal is not only detecting soldering defect in old components but also in new components that differ visually from old components, while maintaining a low leakage rate.
- Taking on the challenge of zero-shot learning, where samples of defective soldering for new components are not present during the training phase.
- The proposal of a hybrid model that incorporates both discriminative and generative models for detecting soldering defects in both old and new components, ensuring the low leakage rate requirement through knowledge exchange and consideration of unknown factors.
- Experiment results on a real-world dataset showing the superiority of the proposed method compared to baseline methods using only discriminative or generative models.

We organize this paper as follows; In Section 2, we discuss related work in existing PCB soldering defect detection methods, unknown awareness in defect detection, compositional zero-shot learning, hybrid generative/discriminative models, and deep metric learning. In section 3, we introduce the proposed model architecture. In section 4, we describe the dataset, evaluation metrics, experiment setup, and experimental results. Finally, in section 5, we summarize our work and discuss future works.

2 RELATED WORK

Model Input and Output: In the field of PCB soldering defect detection using deep learning models, there are two common types of inputs to the model: 1) an image of a PCB board that contains multiple electronic components, and 2) a soldering image of a single electronic component. Our focus in this work is on the latter input type. As shown in Fig. 2, examples of the input images to the model are provided. The output of the model in this work is either "Good," "Bad," or "Unknown." While some related works also classify the different types of defects, we concentrate on reducing the leakage and overkill rates without additional efforts in categorizing the defect types. In this section, we provide a review of related work.

Figure 2: Examples of the input images. 'Good' refers to good soldering. 'Bad' refers to defective soldering. 'Missing,' 'Shift,' 'Stand,' 'Broken,' and 'Short' refers to the defective soldering types.

2.1 EXISTING PCB SOLDERING DEFECT DETECTION METHODS

Wu et al. [2022] proposed a lightweight CNN model called PCBNet capable of locating and classifying the type and defect of an electronic component with low computation complexity while maintaining high accuracy. Liao et al. [2022] proposed ConvNeXt-YOLOX model for solder joint defect detection with high accuracy and speed. Bhattacharya and Cloutier [2022] combined the merits of both transformer [Vaswani et al., 2017] and convolutional networks. All of them focused on balancing speed and accuracy. None of them address the issues encountered in new components.

Ulger et al. [2021] propose a beta-Variational Autoencoders (beta-VAE) architecture for anomaly detection in unrestricted domains with no special lighting and without the existence of error-free reference boards. Instead, we consider where error reference examples of old components are available.

Dai et al. [2020] used a generic deep learning method for both defect localization and classification tasks. For the classification part, an active learning method reduces the labeling workload when an extensive labeled training database is not easily available. On the other hand, our work depends on balancing "knowledge exchange" and "unknown awareness" to achieve the goals – low leakage and overkill rates for new components.

2.2 UNKNOWN AWARENESS IN DEFECT DETECTION

Predictions with low confidence should be considered as unknown. Cheon et al. [2019] applied unknown detection to wafer defect detection tasks in the semiconductor industry. It uses a modified version of K-nearest neighbors (KNN) to determine whether the input belongs to a specific type of defect. When the model cannot determine which type an input belongs to with sufficient confidence, the model claims it to be unknown. The modified KNN, however, is a non-parametric model for which the model should keep the training data in memory.

Habibpour et al. [2021] applied transfer learning methods and uncertainty quantification (UQ) techniques to the casting defect detection task. They believe an uncertainty-aware automatic defect detection solution will reinforce casting production's quality assurance. However, they did not discuss when to say unknown.

Zhou et al. [2021] used a variational autoencoder (VAE) and a Gaussian mixture model (GMM) for the fabric defect detection task. They utilized VAE for feature extraction and image reconstruction and GMM for density estimation. They fitted the GMM with normal data only, which means that the GMM can learn the probability distribution of normal data. Therefore, abnormal samples tend to have a lower probability density than non-defective samples. A threshold can then be determined to distinguish defective and non-defective samples. We also use GMM density estimation in this work and set thresholds for defective, non-defective, and unknown samples.

2.3 COMPOSITIONAL ZERO-SHOT LEARNING

A task similar to ours is compositional zero-shot learning (CZSL), which involves the recognition of the unseen composition of objects (components) and states (defectiveness). In particular, CZSL aims to recognize compositions composed of a set of states and objects. (e.g., red apple, where red is the state and apple is the object). Instead, we focus on recognizing the state of an object, where the object is known in advance or less critical and not interested.

Some CZSL methods [Misra et al., 2017, Purushwalkam et al., 2019, Li et al., 2020, 2022] train two classifiers for state and object, respectively. It is similar to our proposed hybrid classifier, while in our task, the goal is to classify the defectiveness of old and new components with unknown awareness. The task we address here differs from CZSL in the following aspects.

1. *We focus on predicting states only* since predicting objects is generally not critical. With this goal in mind, we can convert a component prediction of the generative model to a defectiveness prediction and do other things, e.g., unknown awareness.
2. *We allow the model not to make any prediction;* when it does make one, the accuracy must approach 100%, thus achieving trustable predictions for real-world applications.
3. *The states in our task are only 'good,' 'bad,' and 'unknown.'* It enables us to effectively share/exchange

knowledge between the discriminative and the generative models.

2.4 HYBRID GENERATIVE AND DISCRIMINATIVE MODELS

Recently, reliable machine learning models have attracted the attention of researchers. A line of research addresses this goal by combining the generative and discriminative models. Grcić et al. [2022], Loh et al. [2022], and Cao and Zhang [2022] applied this idea for anomaly detection, uncertainty capturing, and out-of-distribution detection, respectively. Their successes come from the combination of the strength of these two models. The discriminative models often attain higher predictive accuracy, while the generative ones can deliver reliable predictions.

As in Figure 1, we use GMM models for density (or likelihood) estimation in the generative model. As in Figure 3, we use deep metric learning to make the input z suitable for GMM modeling.

2.5 DEEP METRIC LEARNING

Deep metric learning is often applied to face recognition, person re-identification, and fine-grained image recognition. It enables the model to pull samples of the same class in the embedded space closer and push samples of different classes apart. Its loss functions involve two types: Proxy-based and pair-based.

Proxy-based loss leverages the concept of prototypes so that samples belonging to the same class aggregate in their respective proxy. On the contrary, samples of different classes form separate and independent proxies due to their low similarity, as in [Movshovitz-Attias et al., 2017] [Qian et al., 2019].

Pair-based loss calculates the distances of the paired samples in each mini-batch. The paired samples require more sampling at the training stage [Hadsell et al., 2006, Schroff et al., 2015]. That needs more computation resources than the proxy-based method.

Multi-similarity Loss [Wang et al., 2019] handles the sampling problem by using hard sample mining, which relaxes the sampling problem in the pair-based loss. Furthermore, it penalizes the loss differently by comparing the relationship of anchor, positive and negative, and leads to a performance boost.

We use the multi-similarity loss to train the embedding z of our hybrid architecture, making it suitable for GMM modeling.

3 METHODOLOGY

We train the hybrid defect detector of Figure 1 by procedures depicted in Figure 3. In stage 1, we alternatively train the upper branch for detecting defectiveness and the lower branch for learning cluster embedding z_{com}. In stage 2, we use GMM models to fit z_{com} to realize probabilistic component-type predictions $P(z|y_{com})$. We then convert $P(z|y_{com})$ to the second defectiveness predictions $\hat{y}^2_{def}$. The conversion is optimized using Bayesian optimization with thresholding parameters $\tau 0, \tau 1$, and $\tau 2$. Finally, we combine these two defectiveness predictions to make the unknown-aware final predictions $\hat{y}_{def}$.

3.1 MODEL ARCHITECTURE

We elabrate on the blocks in Figure 1 and Figure 3 as follows:

1. **Pre-trained Feature Extractor** $f_{\theta 1}(\cdot)$. We use the backbone of MobileNetV3 Large pre-trained on Imagenet and remove the original MobileNetV3 Large classification head. Then we adopt the backbone as the feature extractor. $f_{\theta 1}(\cdot)$ maps x to a vector x', $x' = f_{\theta 1}(x) \in \mathcal{R}^{D_{\theta 1}}$. $D_{\theta 1} = 960$.
2. **Shared Encoder** $f_{\theta 2}(\cdot)$. $f_{\theta 2}(\cdot)$ is composed of a fully connected layer. The input dimension of the layer is 960, and the output dimension is 512. $f_{\theta 2}(\cdot)$ maps the extracted features x' to embedding vector z for both the discriminative and the generative models. $z = f_{\theta 2}(x') \in \mathcal{R}^{D_{\theta 2}}$, $D_{\theta 2} = 512$.
3. **Discriminative Model** $\psi(\cdot)$. $\psi(\cdot)$ is a single fully connected layer FC_1 that predicts the defectiveness of a sample. $\psi(\cdot)$ maps z to defect prediction logits $\hat{y}^1_{def} = \psi(z) \in \mathcal{R}^{D_\psi}$, $D_\psi = 2$.
4. **Generative Model** $\gamma(\cdot)$. $\gamma(\cdot)$ is a Gaussian mixture model (GMM) that predicts whether the embedding features z belong to any known component types. The output of the Gaussian mixture model processes is the likelihood of a sample being a specific component type, $P(z|y_{com}) \in \{GMM_1(z), GMM_2(z), ...GMM_T(z)\}$. It will be converted to the second defectiveness prediction $\hat{y}^2_{def}$ later. There are 23 component types in our dataset, so $T = 23$.
5. **Prediction Converter** $\Lambda(\cdot)$. $\Lambda(\cdot)$ converts the likelihood $P(z|y_{com})$ to the second defectiveness prediction $\hat{y}^2_{def} \in \mathcal{R}^{D_\psi}$, $D_\psi = 3$. If $P(z|y_{com})$ is larger than a threshold $h^1_{\cdot,\cdot}$, it is classified as a good sample. If it is smaller than another threshold $h^2_{\cdot,\cdot}$, it is an unknown sample. If it falls between these two thresholds, this sample is a bad sample. There is also a parameter $\tau 0$ for adjusting $P(z|y_{com})$. A detailed discussion can be found in 3.2.

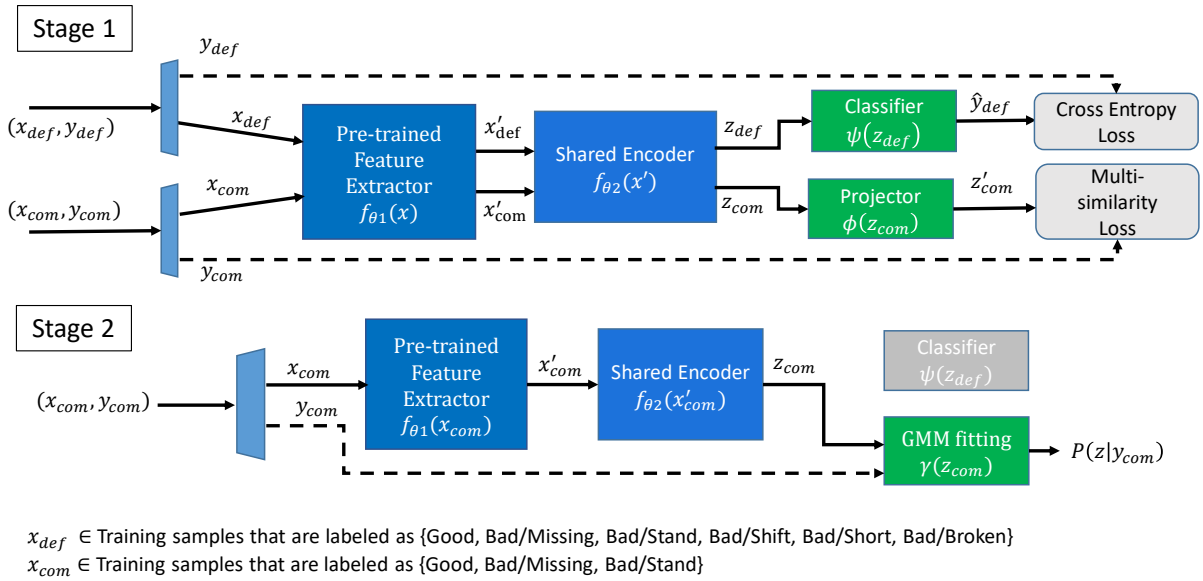


Figure 3: Training procedure of deepGD3 (Hybrid Expert). We train the defect classifier ψ, the component classifier ϕ, f_{θ_1} and f_{θ_2} in stage 1. We then fit the Gaussian mixture models γ in stage 2.

6. **Prediction Combiner** $G(\cdot,\cdot)$. If $\hat{y}^1_{def}$ and $\hat{y}^2_{def}$ do not match or $\hat{y}^2_{def}$ is unknown, we consider the corresponding sample to be unknown. Otherwise, we consider that $\hat{y}^1_{def}$, which is equal to $\hat{y}^2_{def}$, is the final prediction $\hat{y}_{def}$.
7. **Projection Head** $\phi(\cdot)$. $\phi(\cdot)$, in Figure 3, is a fully connected layer FC_2 that uses the multi-similarity loss [Wang et al., 2019] to pull features of good samples with the same component type closer and push features of good samples with different component types away. During stage 1 of the training process, $\phi(\cdot)$ maps z to $z' = \phi(z) \in \mathcal{R}^{D_\phi}$. $D_\phi = 512$. At the end of the training, we discard $\phi(\cdot)$ as Khosla et al. [2020], Chen et al. [2020a,b] did in constrastive learning settings.

3.2 PREDICTION CONVERTER $\Lambda(\cdot)$

There may exist many methods for converting $P(z|y_{com})$ to $\hat{y}^2_{def}$. We introduce a method we found efficient to do that and is stable. The input of our prediction converter Λ are the 23 GMM models $GMM(\mu, \Sigma)$, the embedding z, an adjusting parameter τ_0, and two thresholding paramters τ_1 and τ_2. These three parameters apply to all GMM models and are optimized by Bayesian optimization.

The parameter τ_0 adjusts the covariance matrices of all Gaussians of all GMM models by Equation 1.

$$P_{j,n}(z_i) = GMM_j(z_i; \mu_{j,n}, (\tau_0)^2 \cdot \Sigma_{j,n}), \qquad (1)$$

where $j \in \{1, 2, \ldots, 23\}$ is the number of the component types, n is the number of Gaussians in each GMM model, and i is the sample index. By adjusting the these covariance matrices, the Bayesian optimization find the thresholding parameters τ_1 and τ_2 relatively quickly and resulting better prediction performance.

The two thresholds for all Gaussians of all GMM models are $h^1_{j,n}$ and $h^2_{j,n}$ shown in Equation 2.

$$\begin{aligned} h^1_{j,n} &:= P^1_j(\mu_{j,n}) &&= GMM_j(\mu_{j,n}; \mu_{j,n}, (\tau_1)^2 \cdot \Sigma_{j,n}) \\ h^2_{j,n} &:= P^2_{j,n}(\mu_{j,n}) &&= GMM_j(\mu_{j,n}; \mu_{j,n}, (\tau_2)^2 \cdot \Sigma_{j,n}) \end{aligned} \qquad (2)$$

where $\mu_{j,n}$ indcates the center of a Gaussian. We can therefore adjust all thresholds by adjusting τ_1, and τ_2.

With the adjusted GMM models $P_{j,n}(z_i)$ and these thresholds, the predictoin converters are defned as Eq. 3.

$$\begin{aligned} \hat{y}_{j,n} &:= \begin{cases} \text{good}, & P_{j,n}(z_i) \geq h^1_{j,n} \\ \text{bad}, & h^1_{j,n} > P_{j,n}(z_i) \geq h^2_{j,n} \\ \text{unknown}, & h^2_{j,n} > P_{j,n}(x_i) \end{cases} \\ \hat{y}_{def,j} &:= \begin{cases} \text{good}, & \text{if one of } \hat{y}_{j,n} = \text{good}, \\ \text{bad}, & \text{else if one of } \hat{y}_{j,n} = \text{bad}, \\ \text{unknown}, & \text{else.} \end{cases} \\ \hat{y}^2_{def} &:= \begin{cases} \text{good}, & \text{if one of } \hat{y}_{def,j} = \text{good}, \\ \text{bad}, & \text{else if one of } \hat{y}_{def,j} = \text{bad}, \\ \text{unknown}, & \text{else.} \end{cases} \end{aligned} \qquad (3)$$

Visual explanations of the effect of τ_0, τ_1 and τ_2 are available in the supplementary material. Bayesian optimization then optimizes τ_0, τ_1 and τ_2 to maximize harmonic score H in Equation 5.

3.3 TRAINING PROCEDURE

Figure 3 shows the training procedure of the hybrid generative/discriminative defect detector in Figure 1. We use class-balanced sampling in each mini-batch to deal with the data imbalance issue. The early stop technique is also applied to prevent overfitting.

Our solution trains the upper branch with the defect classifier ψ and the lower branch with the project head ϕ in stage 1. We use the defect type label y_{def} to train the upper branch and the component type label y_{com} to train the lower branch. Cross-entropy loss ℓ_{def} is first computed to update ψ, $f_{\theta 2}$, and $f_{\theta 1}$. Then, the lower branch is trained with multi-similarity loss ℓ_{com} to update ϕ, $f_{\theta 2}$, and $f_{\theta 1}$, enabling knowledge exchange between both branches.

We evaluate the trained models ψ, ϕ, $f_{\theta 2}$, and $f_{\theta 1}$ in the validation set after each epoch. The final model is selected based on the lowest model selection loss, defined as $\ell_\omega = \ell_{com} + \ell_{def}$. After training, Gaussian mixture models are fitted using z_{com} and y_{com} as shown in stage 2 of Figure 3.

A more detailed description of the training procedure, presented as an algorithm, can be found in the supplementary material.

3.4 DETERMINE THE THRESHOLDS BY BAYESIAN OPTIMIZATION

We perform Bayesian optimization using the bayes_opt [Nogueira, 2014–] package on the training and validation sets, with the expected improvement as the acquisition function. The bounds for τ_0, τ_1, τ_2 are given as follows

$$\begin{aligned} &\{\tau_0 | 0 \leq \tau_0 \leq 1.0\} \\ &\{\tau_1 | 0 \leq \tau_1 \leq 1.0\} \\ &\{\tau_1 | 0 \leq \tau_1 \leq 1.0\} \end{aligned} \tag{4}$$

To improve convergence, we shrink the domain around the current optimum using the domain reduction technique. The steps for random exploration are 15 and 25 for Bayesian optimization. We use a harmonic score H to balance the overkill, leakage, and unknown rates, and choose the best combination using Equation 5. The higher the harmonic score, the better the overall performance.

$$H = \frac{1}{3 \times exp(Overkill\ rate)} + \frac{1}{3 \times exp(leakage\ rate)} + \frac{1}{3 \times exp(Unknown\ rate)} \tag{5}$$

Finally, the optimal values of τ_0, τ_1, and τ_2, obtained through Bayesian optimization, are applied to the prediction converter in 3.2, making the proposed hybrid defect detector in Figure 1 fully operational. The complete inference algorithm is included in the supplementary material.

4 EXPERIMENTS AND RESULTS

The proposed method was tested on a dataset from an electronics manufacturing company. The results of the experiment are presented in this section.

4.1 EXPERIMENTAL CONFIGURATION

A subset of 388,702 images was selected from the original dataset, taking into account both data imbalance and simulation speed. There were 23 different component types, and the soldering defect types included missing, shift, stand, broken, and short (as shown in Figure 2). These defect types were consolidated into a single "bad" type. Thus, each sample was annotated with two labels: component type and defect type. The characteristics of the resulting dataset are summarized in Table 1.

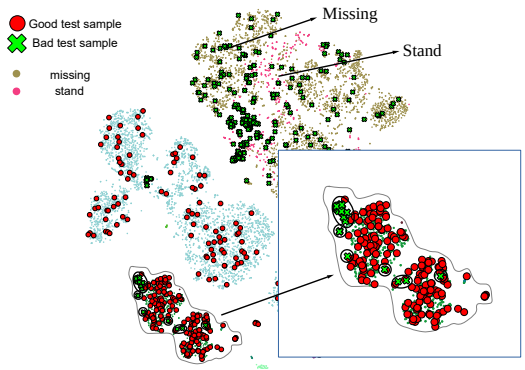
Table 1: Summary for the dataset in use.

Number of images	388,702
Component types	23 types
Defect types	good, bad
Image labels	1 component type, 1 defect type

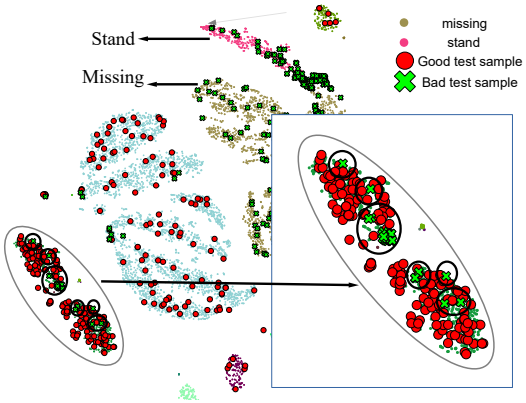
Selection of new and old components: We divided the images into two groups: old components and new components. In the training and validation stage, old components have two labels: component type and defect type, represented as (x, y_{com}, y_{def}). New components only have one label, component type, represented as (x, y_{com}). During the testing stage, both old and new components are used, and the goal is to detect their defectiveness, even though defective new components were not seen in the training and validation stage.

Comparison of different approaches/experts: We compared three approaches: Expert 1 uses a discriminative model and disables the lower branch of Figure.1 in all training/validation/test stages; Expert 2 uses a generative model and disables the upper branch of Figure.1 in all training/validation/test stages; and the Hybrid Expert uses both branches.

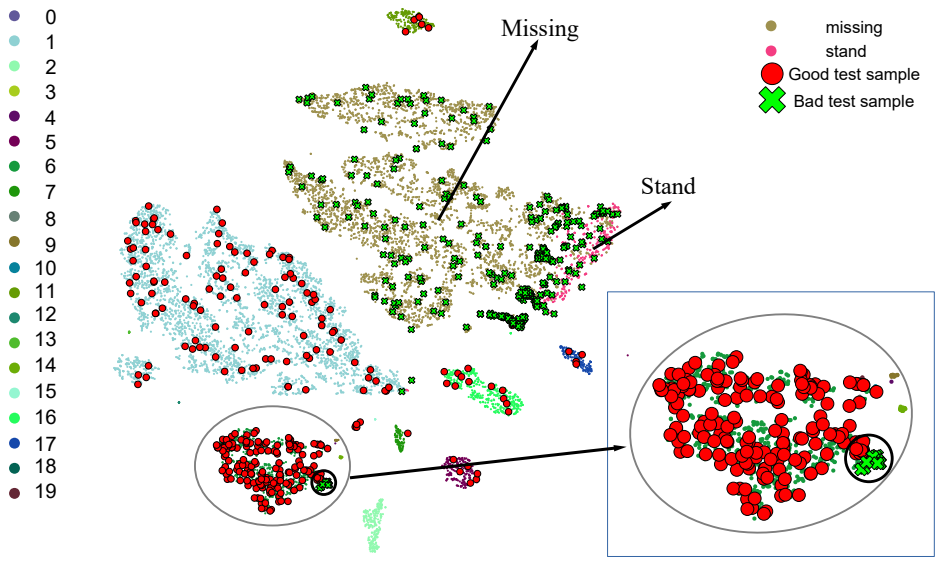
We made 2-D visualizations using t-SNE [Van der Maaten and Hinton, 2008] of the feature embedding z for all three experts. We overlaid the test samples on top of the Good, Missing, Stand training samples to see if good and bad samples are separable under GMM modeling. If a sample is inside the GMM models of good samples, it is considered a good sample. If a sample is inside the GMM models



(a) Test data on top of train data with Expert 1. (Discriminative model). Good test samples and bad test samples are mixed.



(b) Test data on top of train data with Expert 2. (Generative model). Good test samples and bad test samples are mixed.



(c) Test data on top of train data with Hybrid Expert. (Hybrid model). Good test samples and bad test samples are seperable.

Figure 4: 2D visualization of training and test set features. Left: (a) Expert 1, Right: (b) Expert 2, Bottom: (c) Hybrid Expert. In the foreground, red circles indicate good samples in the test set, and green crosses indicate the bad samples of the test set. In the background, each color dot represents a component cluster from the training set.

of Missing, Stand samples, it is considered a bad sample. If a sample is on the boundary of the GMM models of good samples, it may be a bad or unknown sample. If a sample is far from any GMM models, it is considered an unknown sample. Figures 4 (a) and (b) show that, for both Expert 1 and Expert 2, good and bad samples are mixed. However, the Hybrid Expert successfully pushes bad samples to the boundary of the GMM models.

4.2 QUANTITATIVE RESULTS

Evaluation Metrics: Our experimentation evaluations utilize overkill, leakage, and unknown rates as our measurement standards due to their direct relevance to the assembly line needs. These rates are expressed as ratios to the overall number of test samples. The overkill rate represents the ratio of false negative test samples to the overall test samples. The leakage rate displays the ratio of false positive test samples to the overall test samples. The unknown rate indicates the ratio of test samples predicted as unknown to the overall test

samples.

Experiment Results Our results are presented for 1) the entire test samples, including old and new components, 2) the test samples of old components, and 3) the test samples of new components.

Table 2: Comparison of Expert 1, Expert 2, and Hybrid Expert for all test samples.

Method	Overkill (%)	Leakage (%)	Unknown (%)
Expert 1	0.015 ± 0.008	1.827 ± 3.063	-
Expert 2	1.954 ± 0.724	1.942 ± 1.337	0.0 ± 0.0
Hybrid Expert	0.108 ± 0.033	$\mathbf{0.063 \pm 0.075}$	3.7 ± 2.3

Table 2 displays the average overkill and leakage rates for all test samples. The leakage rate of Expert 1 is not up to par. Expert 2 also shows subpar results. Allowing Expert 2 to classify samples as unknown does not improve its performance. Figure 4 (b) shows why this is the case because the bad samples are not on the boundary of GMM models. Nevertheless, the Hybrid Expert performs the best with a low unknown rate of 3.7%.

Table 3: Comparison of Expert 1, Expert 2, Hybrid Expert for old component test samples.

Method	Overkill (%)	Leakage (%)	Unknown (%)
Expert 1	0.017 ± 0.007	0.021 ± 0.011	-
Expert 2	1.282 ± 0.192	2.257 ± 1.495	0.0 ± 0.0
Hybrid Expert	0.129 ± 0.110	$\mathbf{0.019 \pm 0.013}$	3.5 ± 3.0

Table 3 showcases the average overkill and leakage rates for the old component test samples. Expert 1 performs as expected with a favorable leakage rate. Expert 2, however, falls short in comparison. The Hybrid Expert's leakage rate is slightly better than Expert 1.

Table 4: Comparison of Expert 1, Expert 2, Hybrid Expert for new component test samples.

Method	Overkill (%)	Leakage (%)	Unknown (%)
Expert 1	0.010 ± 0.010	3.380 ± 5.540	-
Expert 2	3.739 ± 2.459	0.989 ± 1.713	0.0 ± 0.0
Hybrid Expert	0.126 ± 0.062	$\mathbf{0.090 \pm 0.156}$	3.3 ± 1.6

Table 4 presents the average overkill and leakage rates for the new component test samples. Expert 1 shows a disappointing leakage rate. Expert 2 also fails to meet expectations with a subpar overkill rate. On the other hand, the Hybrid Expert demonstrates the best performance overall.

4.3 ABLATION STUDY

In our studies, we also conducted ablation experiments and found that when 'good' new component samples were not included in the training set, Expert 2 suffered a decline in performance. In contrast, the Hybrid Expert still maintained its overkill and leakage rates, although with a somewhat higher unknown rate. We also examined the effect of the prediction combiner, and these results are available in the supplementary material.

5 CONCLUSION

By leveraging the strengths of a discriminative model (Expert 1) and a generative model (Expert 2), the proposed hybrid defect detector (Hybrid Expert) effectively address the issue of performance degradation when the test samples come from new components for which no defective sample is available.

The hybrid architecture enables the shared encoder network to form a better feature embedding z. The discriminative model makes the first defect prediction $\hat{y}^1_{def}$. The generative model makes probabilistic component prediction $P(z|y_{com})$ by Gaussian mixture models, which determines whether a sample belongs to any known component. The prediction converter converts the probabilistic component prediction to the second defect prediction $\hat{y}^2_{def}$. Finally, a prediction combiner combines the first and second defect predictions to make the final defection prediction $\hat{y}_{def}$. Furthermore, the proposed architecture unknown-aware.

Compared to Expert 1 and Expert 2, the Hybrid Expert reduces the average leakage rate from $1.827\% \pm 3.063\%$ and $1.942\% \pm 1.337\%$ to $0.063\% \pm 0.075\%$ with an unknown rate of $3.706\% \pm 2.270\%$. Our method strikes a balance between overkill, leakage, and unknown rate. The proposed method significantly improves the performance of the new component defect detection task.

The success of our hybrid expert is attributed to three key factors. Firstly, it leverages the knowledge gained from the detection of defects in old components to improve the detection of defects in new components. Secondly, it utilizes a prediction converter to maximize the utilization of the acquired knowledge. Finally, it has the capability to indicate "unknown" when the model's confidence in its predictions is low. These factors contribute to the effectiveness of our hybrid generative/discriminative defect detector and provide a new avenue for further research.

The proposed approach has significant practical value for detecting soldering defects in new components, which is crucial for ensuring the quality and reliability of Printed Circuit Board assemblies. Its potential for application in other scenarios motivates us to continue exploring its capabilities.

References

Abhiroop Bhattacharya and Sylvain G Cloutier. End-to-end deep learning framework for printed circuit board manufacturing defect classification. *Scientific Reports*, 12(1):12559, 2022.

Anna Bosch, Andrew Zisserman, and Xavier Munoz. Scene classification using a hybrid generative/discriminative approach. *IEEE transactions on pattern analysis and machine intelligence*, 30(4):712–727, 2008.

Senqi Cao and Zhongfei Zhang. Deep hybrid models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2022.

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, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.

Sejune Cheon, Hankang Lee, Chang Ouk Kim, and Seok Hyung Lee. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Transactions on Semiconductor Manufacturing*, 32(2):163–170, 2019.

Wenting Dai, Abdul Mujeeb, Marius Erdt, and Alexei Sourin. Soldering defect detection in automatic optical inspection. *Advanced Engineering Informatics*, 43: 101004, 2020.

Akinori Fujino, Naonori Ueda, and Kazumi Saito. A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 764. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 500–517. Springer, 2022.

Maryam Habibpour, Hassan Gharoun, AmirReza Tajally, Afshar Shamsi, Hamzeh Asgharnezhad, Abbas Khosravi, and Saeid Nahavandi. An uncertainty-aware deep learning framework for defect detection in casting products. *arXiv preprint arXiv:2107.11643*, 2021.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673, 2020.

Volodymyr Kuleshov and Stefano Ermon. Deep hybrid models: Bridging discriminative and generative approaches. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, 2017.

Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9335, 2022.

Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11316–11325, 2020.

Shuaidong Liao, Chunyue Huang, Ying Liang, Huaiquan Zhang, and Shoufu Liu. Solder joint defect inspection method based on convnext-yolox. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 12(11):1890–1898, 2022.

Siyuan Brandon Loh, Debaditya Roy, and Basura Fernando. Long-term action forecasting using multi-headed attention-based variational recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2419–2427, 2022.

Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1792–1801, 2017.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python,

2014–. URL https://github.com/fmfn/BayesianOptimization.

Robin Wentao Ouyang, Albert Kai-Sun Wong, Chin-Tau Lea, and Mung Chiang. Indoor location estimation with reduced calibration exploiting unlabeled data via hybrid generative/discriminative learning. *IEEE transactions on mobile computing*, 11(11):1613–1626, 2011.

Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602, 2019.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019.

Rajat Raina, Yirong Shen, Andrew Mccallum, and Andrew Ng. Classification with hybrid generative/discriminative models. *Advances in neural information processing systems*, 16, 2003.

Wolfgang Roth, Robert Peharz, Sebastian Tschiatschek, and Franz Pernkopf. Hybrid generative-discriminative training of gaussian mixture models. *Pattern recognition letters*, 112:131–137, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Furkan Ulger, Seniha Esen Yuksel, and Atila Yilmaz. Anomaly detection for solder joints using β-vae. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(12):2214–2221, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

Hongjin Wu, Ruoshan Lei, and Yibing Peng. Pcbnet: A lightweight convolutional neural network for defect inspection in surface mount technology. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2022.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41 (9):2251–2265, 2018.

Qihong Zhou, Jun Mei, Qian Zhang, Shaozong Wang, and Ge Chen. Semi-supervised fabric defect detection based on image reconstruction and density estimation. *Textile Research Journal*, 91(9-10):962–972, 2021.