CLIP-Vision Guided Few-Shot Metal Surface Defect Recognition

Tianlei Wang, Zeliang Li, Ying Xu, Yikui Zhai, *Senior Member, IEEE*, Xiaofen Xing, *Member, IEEE*, Kailing Guo, *Member, IEEE*, Pasquale Coscia, *Senior Member, IEEE*, Angelo Genovese, *Senior Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, and Fabio Scotti, *Senior Member, IEEE*

***Abstract*—Metal surface defect recognition (MSDR) based on deep learning encounters the challenge of few-shot expert-labeled data. In this study, we proposed a CLIP-vision guided self supervised learning (CVGSSL) framework for representation learning of unlabeled data, completing MSDR using few-shot labeled data. This framework initially generates rich and diverse representation information through multiple CLIP-Vs to ensure effective SSL pretraining, followed by the design of an MLP-adapter to distill knowledge and adapt these representations to recognition tasks. In addition, we constructed a self-constrained loss to address the inherent problem of intraclass and interclass distance ambiguity that causes the representation to fall into an equivocal decision margin. Following label-free pretraining of CVGSSL, the downstream model adapts to one-shot to four-shot defect recognition tasks through fine-tuning. Experimental results demonstrate that CVGSSL outperforms state-of-the-art SSL methods across three public metal surface defect datasets, with the efficacy of the approach validated through extensive ablation experiments.**

***Index Terms*—Contrastive language-image pretraining (CLIP), deep learning, few-shot, self-supervised learning (SSL), surface defect recognition.**

Received 8 November 2024; accepted 24 February 2025. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515011576, in part by the Guangdong Higher Education Innovation and Strengthening School Project under Grant 2020ZDZX3031, Grant 2022ZDZX1032, Grant 2023ZDZX1029, and Grant 2024ZDZX1009, in part by the Wuyi University Hong Kong and Macao Joint Research and Development Fund under Grant 2022WGALH19, in part by the Guangdong Jiangmen Science and Technology Research Project under Grant 2220002000246 and Grant 2023760300070008390, in part by the Jiangmen City Science and Technology Special Commissioner Scientific Research Cooperation Project under Grant 2023760300180008278, and in part by the Guangdong Provincial Key Laboratory of Human Digital Twin under Grant 2022B1212010004. Paper no. TII-24-5880. *(Tianlei Wang, Zeliang Li, and Ying Xu contributed equally to this work.) (Corresponding authors: Yikui Zhai; Kailing Guo; Xiaofen Xing.)*

Tianlei Wang, Ying Xu, and Yikui Zhai are with the Department of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China (e-mail: tianlei.wang@aliyun.com; xuying117@163.com; yikuizhai@163.com).

Zeliang Li is with the School of Future Technologies, South China University of Technology, Guangzhou 510641, China (e-mail: zeliang0li@163.com).

Xiaofen Xing and Kailing Guo are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: xfxing@scut.edu.cn; guokl@scut.edu.cn).

Pasquale Coscia, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti are with the Depatimento di Information, Universita, Degli Studi di Milano, 20133 Milano, Italy (e-mail: pasquale.coscia@unimi.it; angelo.genovese@unimi.it; vincenzo.piuri@unimi.it; fabio.scotti@unimi.it).

Our code is released at https://github.com/deeplearning-LI/CVGSSL.git.

Digital Object Identifier 10.1109/TII.2025.3547353

I. Introduction

METAL surface defect recognition (MSDR) has been a popular research topic in quality inspection of industrial products. MSDR not only ensures the appearance, quality, and durability of products, but also provides accurate feedback for the production process, which is of great significance in the civil field and military field [1].

In recent years, MSDR has transitioned from artificial defect recognition, based on manually extracted features, to intelligent recognition based on deep learning [2]. However, further research has shown that these methods struggle to work under the condition of scarce labels. Meanwhile, labeling specific defects in industrial production requires a significant amount of expert knowledge, which leads to few-shot exploitable labels in defect samples [3]. Several previous works have been explored to tackle the few labeled sample problem in MSDR, including data augmentation using generative adversarial networks (GANs) and metric learning [3], [4]. However, these methods typically necessitate a certain quantity of high-quality labeled data for guidance during the training phase and may not directly leverage existing expert knowledge for the unlabeled data collected during production [5]. Leveraging expert knowledge can alleviate the burden of quality inspection tasks and provide robust support for remedial actions. To this end, it is necessary to explore a representation learning method for unlabeled data so that the data representation can be adapted to the expert empirical knowledge to complete few-shot recognition [6].

Recently, self-supervised learning (SSL) stands out as the most prominent representation learning method for unlabeled data [7]. Specifically, SSL acquires the data clustering capability through unlabeled training, which can be subsequently adapted to downstream tasks via few-shot labeled data. This typically relies on rich and diverse data representations, imbued with dense domain knowledge, to effectively adjust with expert labels [8]. However, metal surface defects (MSDs) are generated with low probability, and these defects do not present clear instance information, such as that found in general datasets (e.g., people, plants, animals), making it challenging to cover

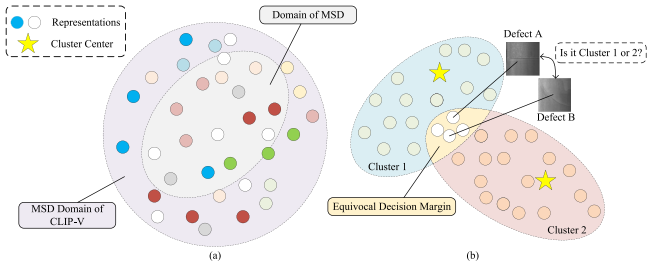


Fig. 1. Diagram of model representation behavior, the dots of the same color represent common features, which may belong to the same category. (a) Indicates that the representation space obtained by CLIP-V for processing MSDs is extremely large. (b) Depicts the equivocal decision margin caused by samples with ambiguity intraclass and interclass distances in MSD.

a sufficiently diverse defect representation space. Furthermore, it constrains the process of fine-tuning (FT) the model with few-shot expert experiential knowledge because the lack of a diverse representation spaces struggle to capture the common features of defect characteristics, making it challenging to achieve effective label-free clustering that can be fine-tuned to align with expert knowledge. Nevertheless, the label-free training nature of SSL makes it a powerful means of addressing the few-shot MSDR problem, as it not only alleviates the burden of expert labeling but also utilizes unlabeled defect data collected during production to align with prior expert experience. Consequently, SSL retains significant potential in few-shot MSDR, provided there is sufficient guidance diversity.

Currently, contrastive language-image pretraining (CLIP) has garnered significant attention for its remarkable instance discrimination capability and its diverse, expansive representation space [9]. Previous studies have utilized CLIP-text and CLIP-vision (CLIP-V) as teacher models to guide student models (downstream models, DMs), thereby endowing the latter with broader domain knowledge for downstream tasks [10]. This is enlightening for achieving efficient few-shot MSDR. However, MSDR often lacks authoritative defect description texts, posing a challenge for integrating existing CLIP-related research. Given the intertwining of the few-shot problem of MSDR and the diverse representation of CLIP-V, an attractive research question arises: is it possible to construct an SSL framework solely through CLIP-V, which acquires simple yet efficient few-shot MSDR? By distilling diverse CLIP-V representations into the DM, it enables the model to acquire a broader range of defect data representation capabilities. However, leveraging CLIP-V representation as a guide for SSL in MSDR still presents several challenges. First, the better representation of CLIP-V is based on vision transformer (ViT) [11], while MSDR models often employ convolutional neural network (CNN) architectures for lightweight models. This results in an inherent representation gap between DM and CLIP-V. Second, as the representation basis of CLIP-V is derived from 400 million image-text pairs, the representation space generated by CLIP-V for defect data is too large to be effectively clustered, as shown in Fig. 1(a).

To address the problems discussed previously, we proposed a novel SSL pretraining approach for MSDs, CLIP-vision guided self-supervised learning (CVGSSL). As shown in Fig. 2, we utilized diverse representations from multiple CLIP-Vs to guide the update of DM, as these varied representations can further enhance the diversity of feature representations [12]. Then, the DM undergoes FT using few-shot labeled samples with expert knowledge to accomplish the few-shot recognition task. Specifically, within the CVGSSL framework, we introduced an **MLP-adapter** to align the representations between CLIP-V and DM, thus mitigating the inherent representation differences between them. Meanwhile, we propose a regularization of diversity constraints, which aims to prevent the representations space of CLIP-V being extremely large. However, due to the inherent ambiguity of intraclass and interclass distances within defect data, the model's decision margin often becomes equivocal [13], as shown in Fig. 1(b). Learning diversified representations through CLIP-V results in more representations deviating from the cluster center and falling within this decision margin. This undoubtedly hampers the DM's ability to learn more robust data feature clustering. To mitigate this problem, we introduced a self-constrained loss function that encourages DM to remain close to their high-level representations, which are abstracted from the **projector**, thereby preventing the model's learned data representation capability from residing within this ambiguous margin. The main contributions of this article are as follows.

1) A novel SSL framework called CVGSSL was proposed, which combines the advantages of diverse characterization of CLIP-V and achieves efficient few-shot MSDR. To the best of our knowledge, this is the first and most cutting-edge SSL framework in MSDR.
2) We proposed an MLP-adapter to adapt the representation of CLIP-V and drive DM to learn these representations. In addition, we regularized the MLP-adapter to prevent marginalization or noise representation from affecting DM.
3) A self-constrained loss was proposed to achieve the constraint goal by enabling DM to learn its high-level representations, thereby preventing the model's representation from falling into the equivocal decision margin of MSD.
4) In our experiments, we evaluated the proposed CVGSSL on three MSD datasets, demonstrating its superiority and effectiveness in few-shot cases. Notably, CVGSSL exhibits stronger performance on imbalanced datasets.

The rest of this article is organized as follows. Section II presents the related work. Section III describes the details of the method. Section IV presents the results of experiments. Finally, Section V concludes this article.

II. Related Works

In this section, we will review the MSDR based on deep learning models in recent years, and briefly introduce SSL and CLIP-based works.

A. Surface Defect Recognition

In the past decade, researchers have discovered that CNN can autonomously and effectively extract image features, eliminating the need for manual feature extraction. For instance, Liu et al. [2] proposed a pyramid network to deal with multisize MSDs. Lian et al. [15] further improved the defect recognition performance by strengthening the generative capacity of model. While these methods can achieve intelligent recognition, they often depend on a large number of expert-labeled samples, posing a challenge for deep learning models dealing with few-shot scenarios. Furthermore, researchers have explored methods to expand existing image databases using image generation techniques to mitigate the few-shot problem. Wen et al. [16] and Gao

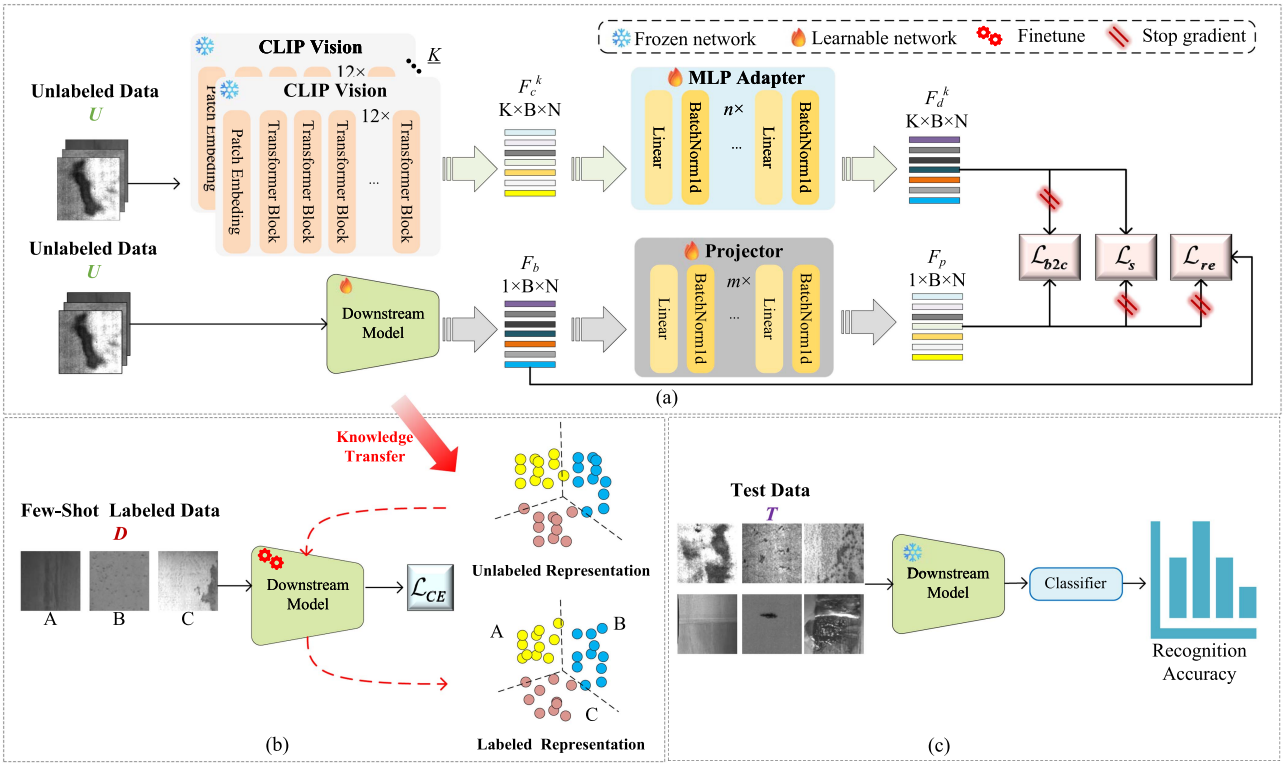


Fig. 2. CVGSSL framework. During the pretraining process, unlabeled data U is used for representation learning of the DM. The learning of various CLIP-V representations by DM is driven by $\mathcal{L}_{b2c}$, while $\mathcal{L}_s$ is employed to constrain the representations learned by DM, preventing excessive divergence. Subsequently, the pretrained DM is fine-tuned using few-shot expert-labeled data D with standard cross-entropy loss ($\mathcal{L}_{\text{CE}}$), ensuring that the representations learned during pretraining align with the actual labels. The performance of the model is then validated using the test data T. Ultimately, a few-shot MSDR based on self-supervised pretraining is implemented. In addition, when calculating losses during the pretraining stage, gradient stopping helps support the model's learning process and prevents instability during backpropagation, particularly when both models are updated simultaneously [14]. (a) Pre-training. (b) Fine-tuning. (c) Testing.

et al. [4] employed GANs to augment surface defect samples and devised an attention module for robust training with additional samples. In addition, researchers have employed metric learning to tackle few-shot tasks for surface defects [17], [18]. However, these methods often overlook the potential of unlabeled data. In reality, unlabeled data generated during production can offer insights similar to past expert experience, facilitating subsequent processing measures. By mining the structural characteristics of unlabeled data, models can acquire reliable data representation capabilities, maximizing the utilization of few-shot expert experience knowledge.

B. Self-Supervised Learning

SSL derives a recognition model suitable for transfer to downstream tasks by learning structured information of unlabeled data. Such a recognition model typically requires only a single labeled sample per class for adaptation to downstream few-shot tasks. SSL typically relies on the rich and diverse structural information provided by the same batch of data to effectively explore the similarities and differences between data, exemplified by MoCov2 [14], SimCLR [19], SimSiam [20]. However, data in MSDR are often scarce and lacks diversity, posing challenges for SSL models. While related works [21], [22] enhance SSL representation learning by enriching unlabeled MSDR data, it is constrained by the original data representation and lacks diversity. By leveraging rich and diverse representations, models can better learn the structural characteristics of data to align with past expert experience knowledge. Thus, the objective of this article is to propose an SSL framework aimed at addressing the challenge of few-shot MSDR by utilizing diverse and rich representation information as guidance.

C. CLIP-Based Works

The CLIP model, trained jointly on language and images, stands as a state-of-the-art (SOTA) deep learning model aimed at achieving high generalization [9]. Through aligning image instances with language descriptions, CLIP enhances the discrimination capabilities of CLIP-V toward instances and enriches the representation within the "vision vocabulary" [23]. CLIP visual representations, based on 400 million text-image pairs, generally exhibit superior data representation capabilities compared to certain models that based on ImageNet, particularly when utilizing models with ViT structures [24]. Some cutting-edge works have successfully accomplished one-shot tasks by harnessing CLIP's remarkable generalization ability [25]. In addition, there are studies that leverage CLIP to enhance the representation of instance information in various tasks [10]. Motivated by these encouraging works, we aim to use the data representation advantages in CLIP-V to guide SSL for few-shot MSDR.

III. Proposed Method

A. Overview

The goal of the CVGSSL framework is to enhance the data representation ability of SSL on unlabeled MSDs to better

respond to few-shot MSDR. One of the significant contributions of this work is the discovery that the diverse characterizations offered by CLIP-V can address the limitations of SSL application in MSDR. However, the primary goal of CVGSSL is for the DM to learn how to maximize the similarity between the representations of identical instances (i.e., the same images processed by different models), which facilitates the discovery of the data structure for MSDs. Specifically, we aim for the representations of the same images, after passing through both DM and CLIP-V, to be as similar as possible. Therefore, we define the expected loss for the process of DM learning from CLIP-V as follows:

$$\mathcal{L}_{b2c} = -\mathbb{E}_B \left[\log \sum_{i=1}^{B} \frac{\exp(S_{ii})}{\sum_{j=1}^{B} \exp(S_{ij})} \right] \tag{1}$$

where $\mathcal{L}_{b2c}$ represents the loss for DM learning the diversity from CLIP-V, the B stands for batchsize, the S_{ii} denotes the element in the ith row and ith column of the similarity matrix S. In addition, we use a function of the form SoftMax$(*)$ to calculate the expectations for each batch. The reason we utilize the SoftMax$(*)$ function to achieve this process is that it aligns with traditional classification tasks, where a negative log-likelihood estimate can be constructed to maximize the similarity between identical instances essentially maximizes the predicted probability of similarity between identical instances [7]. Furthermore, we define the corresponding empirical loss for entire dataset as follows:

$$\hat{\mathcal{L}}_{b2c} = \frac{u}{B} \sum_{l}^{u/B} \left(\log \sum_{i=1}^{B} \frac{\exp(S_{ii})}{\sum_{j=1}^{B} \exp(S_{ij})} \right)_l \tag{2}$$

where u represents the number of samples in the dataset, and $\hat{\mathcal{L}_{b2c}}$ indicates the empirical loss for the entire dataset. However, due to the large representation space of CLIP-V, we introduce an MLP-adapter module and design the $\mathcal{L}_{\text{re}}$ loss to constrain CLIP-V's representation. In addition, we incorporate a projector to project DM's representations into a higher level space, designing the self-constraining regularization $\mathcal{L}_s$ to prevent DM's representations from falling into ambiguous decision boundaries. Finally, add the three $(\mathcal{L}_{b2c}, \mathcal{L}_{\text{re}}, \mathcal{L}_s)$ together to obtain $\mathcal{L}_{\text{total}}$. It is important to note that, for clarity and simplicity, we will use the expected loss as defined in (1) for the subsequent analysis.

For formula 1, the S_{ii} typically exists within S, which is derived from the product of two sets of normalized representations, with one vector required to be transposed to obtain the similarity between the two sets, where S_{ii} represents the diagonal elements of S. Specifically, the similarity matrix S between representations is obtained using the following formula:

$$S = \text{sim}(F_1, F_2) = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1j} \\ S_{21} & S_{22} & \dots & S_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i1} & S_{i2} & \dots & S_{ii} \end{bmatrix} \tag{3}$$

$$\text{sim}(F_1, F_2) = \frac{F_1}{\|F_1\|} \times \frac{F_2}{\|F_2\|} \tag{4}$$

where $F_1 \in \mathbb{R}^{B \times N}$ and $F_2 \in \mathbb{R}^{B \times N}$, which indicate representations of different model outputs from the same batch of images, B and N represent batchsize and dimension of representation. Importantly, the F_2 needs to be transposed for this computation. Both i and j range from 1 to B. In particular, S_{11} through S_{ii} represent the set S^+. In CVGSSL, maximizing S^+ is accomplished by minimizing $\mathcal{L}_{\text{total}}(\theta)$ through the backpropagation mechanism, enabling the θ parameter of the downstream model to learn how to identify similar data representations.

In the remaining contents of this section, we will elucidate the entire framework's implementation process by revealing the principles and motivations behind the CVGSSL design, as shown in Fig. 2. In addition, we will derive how to achieve maximize S^+. Specifically, we will introduce the data forward propagation framework first, encompassing model selection and the acquisition of model inputs and outputs. Second, we will explain the guidance mechanism of CLIP-V and outline the relevant motivations and principles. Next, we will elaborate on how to constrain DM through self-constraint loss. Finally, we will utilize the backpropagation process of CVGSSL pretraining to briefly analyze the impact of $\mathcal{L}_{\text{total}}$ on the model learning process.

B. Data Forward Propagation Framework

SSL often requires ample and diverse samples for successful label-free representation learning, but MSD samples lack this characteristic. Nonetheless, there is an urgent need to grasp the data structure characteristics of unlabeled samples in the production process to match the existing few-shot expert-labeled samples. Consequently, this poses a significant challenge: how to construct a straightforward and efficient SSL implementation for few-shot MSDR.

Motivated by the generalization performance demonstrated by knowledge distillation of SSL and diverse representations of CLIP-V [9], we developed the CVGSSL framework. CVGSSL begins at the diversified representation level of data, focusing on diversity across representations rather than individual instances or samples. Within CVGSSL, we incorporated multiple CLIP-V to process the same batch of images using various data augmentations. This typically introduces implicit representations of more instances [12], further enhancing the rich and diverse representations of CLIP-V. In addition, we constructed an MLP-adapter, composed of n stacked combinations of Linear$(*)$ and BatchNorm1d$(*)$ layers, designed to enhance and adapt these representations. These representations act as a powerful teacher, guiding the DM model as a student. The specific forward process on the teacher side can be articulated as follows:

$$F_c^k = \text{CLIP-V}^k(I^k) \in \mathbb{R}^{K \times B \times N} \tag{5}$$

$$F_d^k = \text{MLP-Adapter}(F_c^k) \in \mathbb{R}^{K \times B \times N} \tag{6}$$

where K, B, and N represent the number of CLIP-V, batchsize, and dimension number of representation, $k \in [1, K]$. F_c^k represents the latent generated by CLIP-V, I^k represents the same batch image of different data enhancement, and F_d^k represents the clustering result of the MLP-adapter on the latent representations. As DM, we selected ResNet18, a neural network of CNN structure, which has reliable inductive bias. Once the DM receives the same batch of images as CLIP-V, it generates its own representation. Subsequently, we construct a projector to enhance these representations further

$$F_b = f(I) \in \mathbb{R}^{1 \times B \times N} \tag{7}$$

$$F_p = \text{Projector}(F_b) \in \mathbb{R}^{1 \times B \times N} \tag{8}$$

where $f(*)$ represents DM. F_b represents the representation of the DM, while F_p denotes its high-level representation. The main

purpose of the projector is to offer the DM a more abstract high-level representation to serve as a learning objective, enabling the DM to learn from diverse representations without losing focus on defect-specific characteristics. Notably, the projector shares a similar structure with the MLP-adapter, consisting of m stacked combinations of Linear$(*)$ and BatchNorm1d$(*)$ layers, is prevalent in SSL models [8]. By performing an operation analogous to clustering on F_b, the representation can be further aggregated to produce more abstract features of F_b. These features help guide the DM toward a more advanced self-representation.

C. Guidance Mechanism of CLIP-V

Traditionally, diverse information is conveyed to DMs through knowledge distillation. Therefore, the diversified representations of CLIP-V, as outlined in Section III-B, can be leveraged to guide the DM in transmitting a broader representation space. This enables DM to cover the data representation domain of MSDR more comprehensively, facilitating better alignment with expert knowledge in few-shot scenarios. However, CLIP-V's diverse representation is derived from the ViT structure, which significantly differs from the CNN structure of DM. Moreover, CLIP-V's representation, trained with 400 million image-text pairs, often encompasses a data representation space that is excessively broad, potentially causing the DM to learn marginalized information. Consequently, we adapt CLIP-V's representation and design the MLP-adapter. We efficiently distill CLIP-V's representation by aligning the DM's output with that of the MLP-adapter.

Initially, this idea is quite intuitive: if the output of DM closely resembles that of the MLP-adapter, its representation space will approximate CLIP-V's, resulting in higher diversity. Therefore, we devise a loss function, $\mathcal{L}_{b2c}$, aimed at aligning the DM representation with the output of the MLP-adapter, facilitating maximal learning of diverse information by the DM. Moreover, owing to the spatial sparsity of defect data and the varying sizes of defect targets, cosine similarity prioritizes variate direction over location, rendering it particularly suitable for defect target recognition. Thus, $\mathcal{L}_{b2c}$ quantifies the relationship between two outputs using cosine similarity. Specifically, the expression for $\mathcal{L}_{b2c}$ is as follows:

$$\mathcal{L}_{b2c} = -\sum_{k=1}^{K} \log\left(\sum_{i=1}^{B} \frac{\exp(\mathrm{sim}(\mathrm{sg}(F_{d,i}^{k}), F_{p,i}))}{\sum_{j=1}^{B}\exp(\mathrm{sim}(\mathrm{sg}(F_{d,i}^{k}), F_{p,j}))}\right) \tag{9}$$

where $\mathrm{sim}(*)$ represents the cosine similarity between two representations. This loss takes the form of SoftMax [14]. In this context, the $\mathrm{sg}(*)$ signifies the cancellation of parameter backpropagation, indicating the stop gradients. $F_{d,i}^{k}$ and $F_{p,i}$ represent the same position in the two representations, otherwise they are different positions. $\mathcal{L}_{b2c}$ aims to make the representations of the same image from different views of F_d^k and F_p similar, promoting the DM to be close to CLIP-V. Besides, F_d^k will cancel the gradient as $\mathcal{L}_{b2c}$ drives the DM, which means that it will only be a constant and serve as a benchmark to guide the model.

Second, due to the broad nature of CLIP-V's representation, certain representations may contribute to marginalized and noisy representations, as show spike peaks of Fig. 3(a)–(c). Thus, we aim to constrain CLIP-V's representation space to better suit

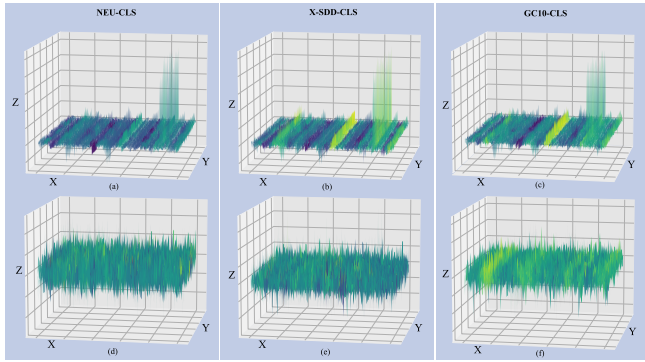


Fig. 3. Representation space diagram. X, Y, and Z indicate the number of samples, dimension number of representation, and representation value, respectively. (a)–(c) Depict the representation of CLIP-V, while (d)–(f) show the representation after $\mathcal{L}_s$ constraints.

defect data. However, direct adjustment of CLIP-V is impractical due to its vast number of parameters. Consequently, we utilize $\mathcal{L}_s$ to guide the adaptation of the MLP-adapter. Specifically, the expression for $\mathcal{L}_s$ is

$$\mathcal{L}_s = -\sum_{k=1}^{K} \log\left(\sum_{i=1}^{B} \frac{\exp(\mathrm{sim}(\mathrm{sg}(F_{p,i}), F_{d,i}^{k}))}{\sum_{j=1}^{B}\exp(\mathrm{sim}(\mathrm{sg}(F_{p,j}), F_{d,i}^{k}))}\right). \tag{10}$$

Notably, F_p cancels the gradient, rendering it equivalent to a constant, thus serving as the target for MLP-adapter learning. $\mathcal{L}_{b2c}$ and $\mathcal{L}_s$ can be seen as adversarial, while $\mathcal{L}_{b2c}$ necessitates the model to learn diverse features, $\mathcal{L}_s$ restricts diversity and enhances relevance to defect data. Besides, we implemented the stop gradient operation to prevent model collapse during training, ensuring a more stable training process [14]. Regardless of whether it is DM learning from CLIP-V or the MLP-adapter learning from DM, one component maintains a fixed learning objective. If gradient stopping is removed, the model may be updated twice using the same output, resulting in unpredictable variations between the initial and subsequent updates. This may lead to a loss of direction in model updates, ultimately destabilizing the learning process.

D. Self-Constrain for Downstream Model

With the guidance of CLIP-V, DM has a decent ability to discriminate instances. However, defects exhibit ambiguity in intraclass and interclass distances, which increasing the likelihood that the diversified representations learned by DM fall within this equivocal decision margin. This highlights a potential drawback of utilizing large, general models for domain-specific tasks. To address this problem, it may be necessary to incorporate more task-specific information, even if it is implicit or abstract. Thus, we develop a self-constrained loss ($\mathcal{L}_{\mathrm{re}}$) aimed at aligning the DM's representation with its abstract, higher level representation to emphasize focus on the task center. By utilizing the $\mathcal{L}_{\mathrm{re}}$, the model is able to reinforce its own output supervision and prevent deviation from the task during CLIP-V guidance. This process resembles human learning, where individuals need to generalize and summarize knowledge after receiving guidance from teachers. In order to minimize deviation during updates, we

use F_p as the target of F_b. It is worth noting that F_p is obtained from the same batch of images, which means that F_b and F_p should be very similar. For this purpose, the $\mathcal{L}_{\text{re}}$ is constructed

$$\mathcal{L}_{re} = -\log\left(\sum_{i=1}^{B}\frac{\exp(\text{sim}(\text{sg}(F_{p,i}), F_{b,i}))}{\sum_{j=1}^{B}\exp(\text{sim}(\text{sg}(F_{p,i}), F_{b,j}))}\right). \quad (11)$$

Through $\mathcal{L}_{\text{re}}$, F_b will approach high-level representation F_p as closely as possible, enhancing DM's ability to find similar representations and thus approaching the cluster centers.

E. Pretraining

Based on the above-mentioned introduction to the construction process of CVGSSL, it can be concluded that CVGSSL is a self-supervised framework extended from CLIP-V. The framework incorporates additional modules, such as the MLP-adapter and projector. $\mathcal{L}_{b2c}$ drives DM to learn from CLIP-V's diverse features adapted through the MLP-adapter, while $\mathcal{L}_s$ constrains the representation space of CLIP-V, ensuring that DM avoids learning excessive noise. $\mathcal{L}_{\text{re}}$ addresses ambiguities in intraclass and interclass distances by guiding DM toward the high-level representations projected by the projector. Simultaneously, CVGSSL trains the DM on unlabeled data and fine-tunes it using a few-shot labeled dataset. To achieve the objective of CVGSSL, $\mathcal{L}_{b2c}$, $\mathcal{L}_s$, and $\mathcal{L}_{\text{re}}$ are directly added without assigning specific weights to control the importance of each loss

$$\mathcal{L}_{\text{total}} = \frac{1}{K}(\mathcal{L}_{b2c} + \mathcal{L}_s) + \mathcal{L}_{\text{re}}. \quad (12)$$

Based on $\mathcal{L}_{\text{total}}$, we further explore the influence of $\mathcal{L}_{\text{total}}$ on model update during training through the backpropagation mechanism. Obviously, $\mathcal{L}_{b2c}$, $\mathcal{L}_s$, and $\mathcal{L}_{\text{re}}$ share similar structures, all focusing on instance similarity and utilizing the SoftMax$(*)$ function. Thus, to facilitate an intuitive and concise analysis of the $\mathcal{L}_{\text{total}}$ process, we focus solely on $\mathcal{L}_{b2c}$ and omit the summation symbols except for the SoftMax$(*)$ function

$$\mathcal{L}'_{b2c} = -\log\left(\frac{\exp(S_{ii})}{\exp(S_{ii}) + \sum_{j=1,j\neq i}^{B} S_{ij}}\right). \quad (13)$$

When comparing two groups of images with the same image but different views, the calculation of similarity primarily focuses on the diagonal of S. The backpropagation and gradient analysis as carried out as follows:

$$\frac{\partial \mathcal{L}'_{b2c}}{\partial S_{ii}} = \frac{\exp(S_{ii})}{\exp(S_{ii}) + \sum_{j=1,j\neq i}^{B} S_{ij}} - 1 \quad (14)$$

$$\theta_{\text{new}} = \theta_{\text{old}} - \text{LR} \times \nabla\theta, \nabla\theta = \frac{\partial \mathcal{L}'_{b2c}}{\partial S_{ii}} \quad (15)$$

where θ represents the DM parameters, LR represents learning rate. The goal of the gradient descent method's update rule is to minimize the loss function $\mathcal{L}$. Therefore, it is important to ensure that S_{ii} has a gradient value as close to 0 as possible in order to achieve model convergence. In the case of $\mathcal{L}_{b2c}$, $\mathcal{L}_s$, and $\mathcal{L}_{\text{re}}$, CVGSSL utilizes $\mathcal{L}_{b2c}$ to enhance DM's ability to learn instance discrimination and diverse instance information from CLIP-V. $\mathcal{L}_s$ aims to maintain proximity between the MLP-adapter and DM, while mitigating excessive representation space and reducing unstable representation offset. $\mathcal{L}_{\text{re}}$ enhances DM's focus on the clustering center to prevent representations from falling into equivocal margin. These processes enable CVGSSL to construct a robust model for transfer to few-shot MSDR during the pretraining phase.

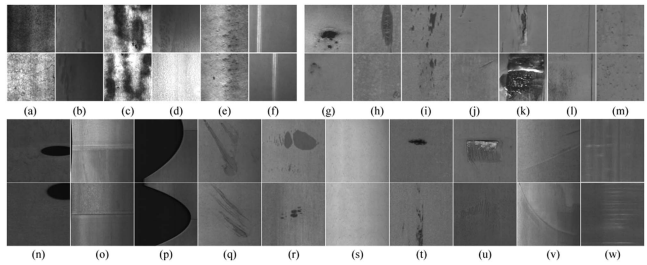


Fig. 4. Examples of datasets. (a)–(f), (g)–(m), and (n)–(w) belong to NEU-CLS dataset, X-SDD-CLS dataset, and GC10-CLS dataset, respectively.

IV. Experiments

A. Task Setup

1) Datasets: In order to verify the performance of few-shot MSDR of the proposed CVGSSL, we selected the public and widely studied NEU-CLS, X-SDD-CLS, and GC10-CLS datasets [22]. The NEU-CLS dataset is a uniform dataset with a total of six defects, each of which contains 300 image samples. The X-SDD-CLS dataset is a nonuniform dataset, which contains seven kinds of defects, and the seven defect types have sample counts of 203, 122, 63, 203, 397, 238, and 134. The GC10-CLS dataset is a nonuniform dataset, with a total of ten classes. The ten classes have sample counts of 329, 513, 265, 354, 569, 884, 347, 85, 74, 143. See Fig. 4 for details. The 60% of NEU-CLS and GC10-CLS was randomly selected to constitute the unlabeled training set $U = \{x_i\}_{i=1}^{u}$, while 70% of X-SDD-CLS for U, consistent with previous work [22]. In addition, a few-shot dataset $D = \{(x_i, y_i)\}_{i=1}^{s}$ was extracted from U to facilitate FT of the model and conduct a comprehensive evaluation. D comprises few-shot expert-label information. Moreover, the remaining samples were built for the test set $T = \{(x_i, y_i)\}_{i=1}^{z}$. The variables u, s, and z denote the quantities of unlabeled samples, few-shot samples, and test samples, respectively. In addition, both the unlabeled training set U and the test set T remain identical across all methods. However, since the dataset D is randomly sampled from U and contains only one to four labeled samples per class, we will sample D 100 times and conduct 100 FT experiments.

2) Evaluation Metric: SSL evaluates model performance by FT and LE [8]. FT adjusts the parameters of the entire model to better suit downstream tasks, and the performance of FT is a strong indicator of the model's generalization strength obtained during SSL pretraining. Linear evaluation (LE) freezes the backbone of the model and only updates the classifier layers, which determines whether the model has learned task-relevant capabilities. The model's performance is presented in terms of accuracy $\pm$ standard deviation, where accuracy (acc) reflects the overall performance of the model and standard deviation indicates how the model's performance fluctuates.

3) Implementation Details: CVGSSL employs a batch size of 64 and conducts pretraining for 300 iterations to obtain the final weights for FT and LE. The SGD optimizer is utilized in CVGSSL pretraining, as well as in the downstream models

TABLE I
FT AND LE COMPARISON WITH SOTA ALGORITHM BASED ON NEU-CLS DATASET

Mode	FT				LE			
Methods	One-shot	Two-shot	Three-shot	Four-shot	One-shot	Two-shot	Three-shot	Four-shot
Rotation	57.59±6.67	68.40±5.95	73.83±4.97	77.78±4.59	33.64±3.99	37.40±3.97	40.69±4.38	43.38±4.00
InstFeat	56.40±7.30	70.67±6.58	78.00±5.30	82.64±4.21	46.75±6.28	56.89±5.34	62.84±4.55	67.52±4.93
MoCov2	74.67±8.28	83.05±5.01	86.48±3.82	88.52±3.48	74.49±7.35	84.43±4.82	88.20±3.71	90.37±2.57
SimCLR	75.02±7.42	85.22±4.60	88.71±4.16	91.29±3.29	74.33±7.53	85.46±4.83	88.83±4.13	91.24±3.03
Fixmatch	39.45±8.08	78.49±4.71	83.95±4.65	91.92±2.67	30.51±6.23	75.20±4.44	82.66±4.44	92.09±2.08
SwAV	48.61±7.73	59.72±7.36	66.69±6.85	71.02±5.82	48.67±7.29	58.99±6.31	64.47±5.01	68.60±4.34
BYOL	56.37±9.87	71.83±8.39	78.58±6.54	82.44±5.21	63.52±7.68	78.31±6.49	84.44±4.38	87.35±3.73
SimSiam	49.00±9.00	62.76±7.38	70.38±5.60	73.91±5.66	50.35±8.80	63.75±7.51	71.34±5.55	75.30±5.03
MPL	51.83±9.97	52.58±7.38	81.79±5.61	86.00±3.61	52.23±9.75	50.14±7.13	85.28±3.92	89.16±2.35
SimMatch	79.43±6.67	90.87±4.17	92.38±4.97	93.11±2.11	80.05±5.44	91.38±4.14	92.77±2.12	92.48±1.86
FDCL	91.43±6.04	95.35±2.90	96.04±1.88	96.48±1.42	91.43±6.04	95.35±2.90	96.05±1.87	96.47±1.42
TSMMIS	**93.77±3.59**	94.27±2.16	95.20±1.55	95.51±1.37	**93.77±5.68**	94.56±2.39	95.33±2.69	95.78±1.71
CVGSSL	92.21±5.00$^{\downarrow 1.56}_{(acc)}$	**96.05±2.43**$^{\uparrow 0.7}_{(acc)}$	**97.09±0.86**$^{\uparrow 1.05}_{(acc)}$	**97.41±0.86**$^{\uparrow 0.93}_{(acc)}$	92.77±5.86$^{\downarrow 1.00}_{(acc)}$	**95.98±2.10**$^{\uparrow 0.63}_{(acc)}$	**97.19±1.11**$^{\uparrow 1.14}_{(acc)}$	**97.49±1.02**$^{\uparrow 1.02}_{(acc)}$

TABLE II
FT AND LE COMPARISON WITH SOTA ALGORITHM BASED ON X-SDD-CLS DATASET

Mode	FT				LE			
Methods	One-shot	Two-shot	Three-shot	Four-shot	One-shot	Two-shot	Three-shot	Four-shot
Rotation	52.85±8.50	64.78±5.53	71.32±4.55	75.96±3.87	31.66±7.94	39.73±7.31	44.67±6.99	48.36±5.64
InstFeat	44.10±6.28	57.27±5.69	65.86±4.91	71.18±4.90	40.77±6.25	57.80±5.66	65.65±4.49	70.67±4.42
MoCov2	48.23±7.50	59.30±6.82	64.81±5.74	68.00±5.06	45.67±7.21	61.50±6.85	68.38±5.15	72.31±5.18
SimCLR	46.93±7.04	62.32±6.40	70.52±5.30	75.59±4.95	48.20±7.62	59.07±6.85	64.94±5.92	67.63±5.23
FixMatch	41.70±9.40	61.99±5.79	80.09±4.53	87.02±2.51	38.30±9.54	59.52±5.88	79.60±4.43	87.19±2.40
SwAV	41.55±7.17	52.09±6.46	59.82±5.34	64.23±5.37	40.29±7.24	50.30±6.25	55.88±5.60	60.02±5.47
BYOL	43.00±7.04	57.28±6.70	66.64±6.66	71.76±5.87	43.57±7.05	60.36±6.43	68.55±5.27	73.80±5.27
SimSiam	31.56±7.30	40.17±6.22	44.11±6.78	47.88±8.00	31.13±7.39	40.24±6.67	44.39±5.21	54.38±4.47
MPL	49.28±8.54	65.75±5.07	77.96±4.00	82.42±4.09	46.59±8.59	67.53±5.23	79.96±3.77	80.71±3.98
SimMatch	49.75±9.06	65.74±6.16	76.27±5.54	81.84±4.14	49.19±8.48	65.80±6.49	75.48±6.12	78.08±3.81
FDCL	71.42±8.60	81.59±5.55	87.37±3.36	88.46±3.70	71.42±8.62	81.58±5.57	86.40±4.08	88.48±3.70
TSMMIS	72.68±9.87	78.93±6.48	83.22±2.89	84.62±1.97	73.89±7.77	79.17±6.89	84.00±2.27	84.37±2.32
CVGSSL	**81.70±7.89**$^{\uparrow 9.02}_{(acc)}$	**89.39±4.16**$^{\uparrow 7.8}_{(acc)}$	**91.23±3.00**$^{\uparrow 3.86}_{(acc)}$	**92.27±2.23**$^{\uparrow 3.81}_{(acc)}$	**82.25±6.84**$^{\uparrow 8.36}_{(acc)}$	**89.38±3.83**$^{\uparrow 7.8}_{(acc)}$	**91.55±2.75**$^{\uparrow 5.15}_{(acc)}$	**92.52±2.24**$^{\uparrow 4.04}_{(acc)}$

FT and LE. The experiments are conducted on an RTX2080Ti 11 G GPU, utilizing the Python programming language within the PyTorch framework. To ensure reliability and performance stability, we performed 100 experiments on both FT and LE, averaging the results across one-shot to four-shot scenarios.

B. Performance Comparison With SOTA SSL

The SOTA methods include well-known approaches, such as Rotation [26], InstFeat [27], MoCov2 [14], SimCLR [19], FixMatch [28], SWAV [29], BYOL [30], SimSiam [20], MPL [31], and SimMatch [32]. Furthermore, FDCL [21] and TSMMIS [22] are introduced, which are the SSL methods of MSDR. These methods learn data representations through pretraining and transfer them to downstream tasks for FT.

1) Recognition on NEU-CLS Dataset: CVGSSL exhibits superior performance in one-shot to four-shot scenarios for both FT and LE, as illustrated in Table I. However, in the 1-shot case, FT and LE are lower than TSMMIS by 1.56% and 1.00%, respectively. This outcome is expected. CVGSSL guides DM solely based on the representation of the not updated CLIP-V. While this approach tends to introduce more diverse instance features, it may also introduce additional noise. In the one-shot case, the model contends with limited supervision information to adjust for noise effects. Nonetheless, CVGSSL excels when sufficient supervision information is available, indicating that increased supervision mitigates noise introduced by diversity. In addition, MoCov2, SimCLR, and other methods perform poorly due to the limited number of instance representations in pretraining of MSDR. Even GAN-based FDCL is impacted by the uneven quality of generated data. These findings demonstrate CVGSSL's effective utilization of CLIP-V's diversified representation for labelless self-supervised training, yielding a reliable recognition model for downstream tasks with robust recognition patterns.

2) Recognition on X-SDD-CLS Dataset: X-SDD-CLS poses greater challenges due to a more extensive range of categories and an imbalanced distribution among them. Despite this difficulty, CVGSSL consistently outperforms other methods in the one-shot to four-shot scenarios, as illustrated in the Table II. Notably, in one-shot scenarios, CVGSSL surpasses the next best methods in FT and LE by 9.02% and 8.36%, respectively. Despite the long-tail and nonuniform distribution of data, CVGSSL demonstrates unexpected performance. Diverse instance representations facilitate a deeper understanding of differences between instances, effectively addressing the imbalanced data representation challenge. In addition, compared to the most accurate one-shot to four-shot method, CVGSSL can achieve a lower standard deviation.

3) Recognition on GC10-CLS Dataset: The GC10-CLS dataset stands out as the most challenging due to its increased class count and pronounced class imbalance. Nonetheless, CVGSSL consistently attains optimal performance across the one-shot to four-shot scenarios, as illustrated in the Table III. In FT, CVGSSL achieves performances of 61.65%, 71.78%, 74.08%, and 76.13% for the one-shot to four-shot cases,

TABLE III
FT AND LE COMPARISON WITH SOTA ALGORITHM BASED ON GC10-CLS DATASET

Mode	FT				LE			
Methods	One-shot	Two-shot	Three-shot	Four-shot	One-shot	Two-shot	Three-shot	Four-shot
Rotation	32.08±4.83	42.55±4.06	48.20±4.57	52.58±4.20	21.85±3.99	25.54±3.68	27.40±3.15	29.52±2.66
InstFeat	38.39±5.98	47.81±5.26	54.82±5.02	58.78±4.58	36.91±5.82	48.00±4.98	55.99±4.48	59.94±3.95
MoCov2	50.73±5.52	58.72±4.75	62.81±4.60	66.17±3.99	52.98±5.16	61.09±4.98	65.69±4.80	68.84±4.15
SimCLR	43.67±6.44	53.53±5.63	59.01±3.95	61.85±3.70	43.76±6.43	53.37±5.78	58.80±4.19	61.77±3.66
FixMatch	26.46±3.63	33.85±3.60	49.15±5.08	54.83±3.93	24.97±3.34	29.99±2.81	49.78±4.42	54.34±3.84
SwAV	36.06+6.70	44.13±6.07	47.88±5.42	51.50±3.85	36.59±5.90	43.96±5.41	47.92±4.14	51.19±3.66
BYOL	37.60±6.08	47.77±5.68	55.17±5.83	60.13±5.23	45.31±5.93	57.83±5.34	65.43±4.53	69.71±4.32
SimSiam	21.91±2.32	24.41±2.45	25.26±2.76	25.58±2.72	21.87±2.81	24.15±2.78	25.32±2.59	26.23±2.61
MPL	28.56±4.52	48.85±5.07	57.15±4.03	60.47±4.49	28.65±5.25	48.07±4.25	58.41±4.15	61.65±4.47
SimMatch	44.81±6.45	57.37±5.01	61.72±3.88	71.68±4.03	43.97±6.99	57.49±4.84	60.53±4.17	72.81±3.87
FDCL	58.47±5.35	65.81±4.79	70.66±4.29	73.40±3.78	58.47±5.35	65.82±4.80	70.69±4.29	73.43±3.78
TSMMIS	58.50±5.20	67.10±5.05	72.51±5.25	74.24±5.01	58.49±5.79	67.03±5.93	73.47±4.17	74.65±4.46
CVGSSL	**61.65±5.23** $^{\uparrow 3.18}_{(acc)}$	**71.78±4.98** $^{\uparrow 4.68}_{(acc)}$	**74.08±4.21** $^{\uparrow 1.57}_{(acc)}$	**76.13±3.74** $^{\uparrow 1.89}_{(acc)}$	**61.82±5.41** $^{\uparrow 3.33}_{(acc)}$	**70.53±4.91** $^{\uparrow 3.5}_{(acc)}$	**74.23±4.49** $^{\uparrow 0.76}_{(acc)}$	**75.47±3.99** $^{\uparrow 0.82}_{(acc)}$

respectively. Similarly, in LE, CVGSSL demonstrates performances of 61.82%, 70.53%, 74.23%, and 75.47%, respectively, while maintaining a relatively low standard deviation.

The experimental results demonstrate that CVGSSL exhibits superior representation learning capabilities for unlabeled defect data when compared to other SSL methods. CVGSSL introduces diverse representations derived from the visual-language pretrained CLIP-V, enabling the DM to encompass a broad spectrum of representations, thereby facilitating adaptation to few-shot expert-labeled data. Particularly noteworthy is CVGSSL's superior performance in handling imbalanced samples, attributed to the augmented diversity in instance representations, which expands sample coverage and bolsters the model's generalization capabilities. Moreover, the heightened richness of features contributes to an enhanced capacity for discerning and capturing distinctions between features. Furthermore, the incorporation of $\mathcal{L}_{\text{re}}$ in CVGSSL directs the model to explore diversity in a contextually relevant manner, mitigating the risk of the model becoming disoriented during label-free training. To this end, CVGSSL attains the SOTA few-shot MSDR capability.

C. Ablation Results and Discussion

In our ablation experiment, we first discussed the effects of the selected hyperparameters K and n, as well as the choice of CLIP-V and DM model structures, on model performance. In addition, we assessed the impact of the projector level on performance. Furthermore, we investigate the impact of $\mathcal{L}_{b2c}$, $\mathcal{L}_s$, $\mathcal{L}_{\text{re}}$ on the training process of CVGSSL. Unless otherwise specified, all experiments were conducted on the category-balanced datasets NEU-CLS and the category-imbalanced X-SDD-CLS.

1) Ablation of Key Factors: Regarding the choice of K and n, we examined their impact on model performance under various CLIP-V structures (e.g., ViTB16 and RN50) and DM architectures (e.g., ViTB16 and RN18) in both one-shot and four-shot conditions. We set the number of MLP-adapter layers to $n = 3$ to align with the layer count typical in general SSL models. Notably, this MLP structure in CVGSSL is designed to accommodate the characterization of CLIP-V rather than to project or predict specific representations. Our guideline for determining optimal values for K and n is as follows: A parameter that is too small may fail to achieve the desired effects, whereas one that is too large risks overfitting. As illustrated in Fig. 5(a)–(b), increasing K results in greater performance variance for DM models based on the ViT structure, particularly for the nonuniform dataset X-SDD-CLS (yellow curve). When RN18 is used as the downstream model (blue and red curves), selecting ViTB16 as CLIP-V yields better performance. Here, model performance initially improves with increasing K before declining, with the optimal value identified at $K = 4$. Moreover, while an increased number of instances can enhance learning outcomes, which is due to richer features of multiple enhanced views [12], an excessive quantity may introduce unavoidable noise. Fig. 5(c)–(d) demonstrates that when $K = 4$ is fixed, variations in n affect performance similarly. The optimal scenario occurs with ViTB16 as CLIP-V and RN18 as DM, where the ideal n value is 3, particularly for nonuniform datasets. Notably, while the choice of K and n values is influenced by the structures of CLIP-V and DM, the best performance occurs when ViTB16 is selected as CLIP-V and RN18 as DM. This highlights the strength of the CVGSSL framework, which effectively narrows the differences between various model structures. In summary, the selection criteria for hyperparameters K and n lead us to choose $n = 3$ and $K = 4$ as the optimal configuration, considering the performance across both uniform and nonuniform datasets.

Then, to investigate the impact of different CLIP-V structures on DM, we adopted two structural systems: the ViT series and the RN series. At this stage, we set $K = 4$ and $n = 3$ for the experiment. From Fig. 5(e)–(f), it is evident that the ResNet structure models of RN50 and RN101 exhibit lower pretraining effects on DM compared to the ViT structure, particularly evident in the nonuniform dataset X-SDD-CLS. This suggests that the ViT structure of CLIP-V offers more diverse characterizations, attributed to its enhanced focus on global features. Particularly in the one-shot scenario, the model guided by ViT-B16 demonstrates a recognition performance 8.88% higher than the model guided by RN50. This underscores the necessity of employing the ViT structure for CLIP-V.

Furthermore, we examined DMs with varying parameter counts and architectures in the NEU-CLS dataset. In this context, we utilized ViT-B16 as CLIP-V. Fig. 5(g) illustrates that ResNet18 outperforms ViT-based DMs, suggesting that CNN structures are more suitable for industrial data, particularly metal defect data. This also depends on the properties of the defective data. Defects tend to exist in local space, and the CNN structure exhibits a stronger inductive bias toward local features compared to the ViT structure. In addition, lightweight models, whether ViT or ResNet, tend to exhibit better performance on defect data due to their parameter-data match, preventing overfitting. Based on the parameters $K = 4, n = 3$, CLIP-V = ViTB16, and

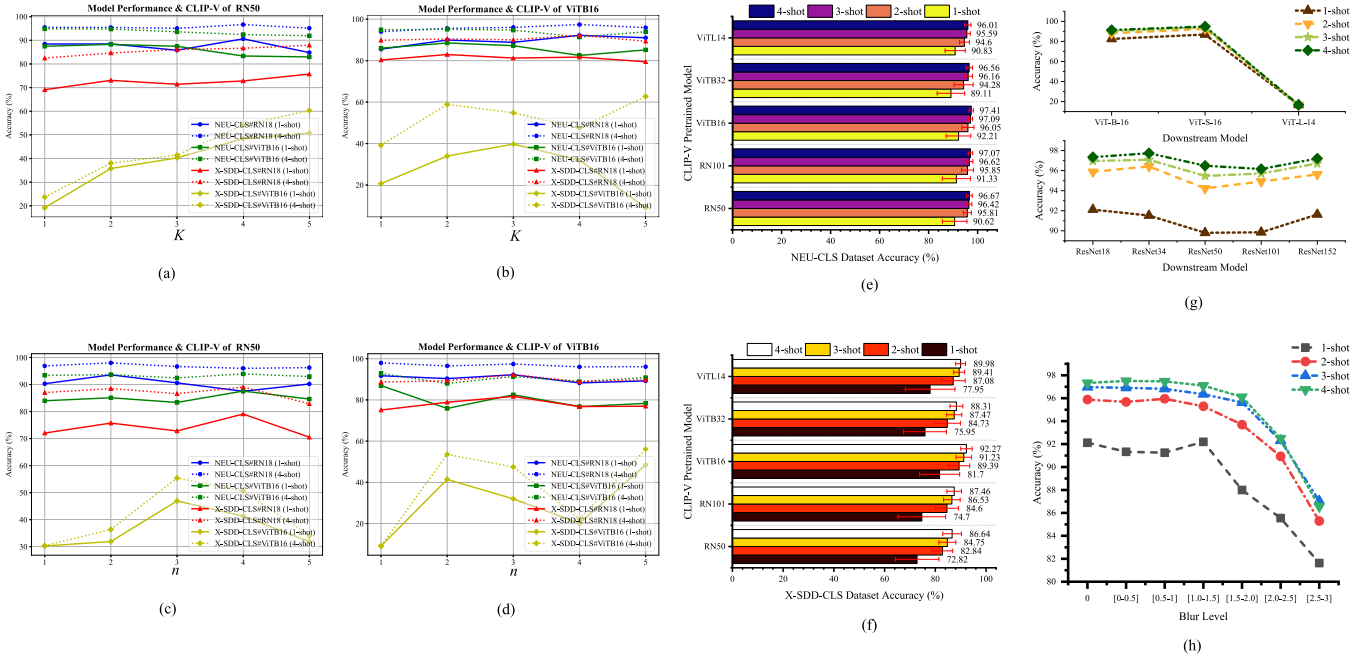


Fig. 5. Influence of key factors. (a)–(b) Are obtained when $n = 3$, while (c)–(d) are obtained when $K = 4$, which illustrate the relationship between the values of K and n with the CLIP-V structure (ViTB16, RN50), downstream model architectures (ViTB16, RN18), and the datasets (NEU-CLS, X-SDD-CLS). where "NEU-CLS#RN18(1-shot)" denotes the one-shot performance curve for DM RN18 in the NEU-CLS dataset. (e)–(f) Detail the performance trend of various CLIP-V structures when the DM is RN18. (g) Illustrates the performance variation across different DMs and their respective parameter counts. (h) Demonstrates the robustness of the model when addressing fuzzy samples.

TABLE IV
ACCURACY IN ONE-SHOT TO FOUR-SHOT SETTINGS FOR DATASETS AND PROJECTOR

DATASETS	m	One-shot	Two-shot	Three-shot	Four-shot
NEU-CLS	0	88.48±6.23	94.31±2.65	96.29±1.36	96.85±1.00
	1	85.54±6.31	91.71±3.89	94.06±2.81	94.97±1.62
	2	**92.21±5.00**	**96.05±2.43**	**97.09±0.86**	**97.41±0.86**
	3	89.48±5.84	93.99±2.52	95.80±1.28	96.25±1.18
	4	85.67±6.69	92.06±3.48	93.22±2.55	93.93±1.70
	5	88.50±5.12	92.75±2.97	94.37±1.82	95.13±1.64
X-SDD-CLS	0	75.33±7.68	84.67±6.00	88.39±2.78	89.86±2.29
	1	74.50±9.31	78.98±8.38	84.49±2.99	86.33±2.19
	2	**81.70±7.89**	**89.39±4.16**	**91.23±3.00**	**92.27±2.23**
	3	79.56±8.86	87.66±4.81	89.87±2.24	90.24±1.61
	4	79.51±9.90	88.56±2.65	89.93±2.28	90.31±1.90
	5	76.70±9.45	86.18±4.81	88.36±1.94	89.42±2.35

TABLE V
ACCURACY IN ONE-SHOT TO FOUR-SHOT SETTINGS FOR DIFFERENT DATASETS AND MODES

DATASETS	Mode	One-shot	Two-shot	Three-shot	Four-shot
NEU-CLS	None	71.57±7.15	80.87±6.11	85.83±3.73	88.42±2.73
	$\mathcal{L}_{b2c}$ ↑	74.16±8.21	83.76±5.82	88.92±3.67	90.49±2.80
	$\mathcal{L}_{b2c} + \mathcal{L}_s$ ↑	88.48±6.23	94.31±2.65	96.29±1.36	96.85±1.00
	$\mathcal{L}_{\text{total}}$↑	**92.21±5.00**	**96.05±2.43**	**97.09±0.86**	**97.41±0.86**
X-SDD-CLS	None	61.50±10.22	73.17±7.69	79.34±4.35	82.01±3.81
	$\mathcal{L}_{b2c}$ ↑	62.76±9.60	75.52±4.93	78.77±4.32	81.92±3.49
	$\mathcal{L}_{b2c} + \mathcal{L}_s$ ↑	75.33±7.68	84.67±6.00	88.39±2.78	89.86±2.29
	$\mathcal{L}_{\text{total}}$↑	**81.70±7.89**	**89.39±4.16**	**91.23±3.00**	**92.27±2.23**
GC10-CLS	None	41.78±5.60	49.32±4.54	52.48±3.99	53.81±3.88
	$\mathcal{L}_{b2c}$	41.53±5.45	46.64±4.82	48.46±3.56	49.71±3.54
	$\mathcal{L}_{b2c} + \mathcal{L}_s$ ↑	57.69±5.92	62.75±4.37	64.59±3.72	65.29±3.38
	$\mathcal{L}_{\text{total}}$↑	**61.82±5.41**	**70.53±4.91**	**74.23±4.49**	**75.47±3.99**

DM = ResNet18, we test the robustness of the model against fuzzy samples in this scenario. Fig. 5(h) indicates that the DM pretrained by CVGSSL exhibits robustness within the blur level [0–1.5]. However, there is a notable performance drop when identifying more ambiguous samples. Nevertheless, the CVGSSL pretrained DM demonstrates strong adaptability to sampling blur induced by machine jitter and other conditions in industrial production.

Regarding the parameter m, the projector is a common module in self-supervised models, and exploring its layers can improve the projection of DM representations into higher level representations. As shown Table IV, the optimal performance is achieved when the projector is set to $m = 2$. Regardless of whether it is for one-shot or four-shot settings, the performance initially increases and then decreases as the number of layers grows. This is because a shallow projector struggles to effectively project the higher level representations of the DM, leading to poor learning, while a deeper projector carries the risk of overfitting.

Based on the above-mentioned analysis, we can identify the optimal key factors. When $K = 4$, $n = 3$, $m = 2$, and the DM selects ResNet18 while the CLIP-V pretraining structure chooses ViTB16, the DM can achieve optimal few-shot recognition performance.

2) Ablation of Loss Function: We first conduct a *quantitative* analysis of the effects of different loss functions. From Table V, it is evident that the model has already achieved certain performance without any additional special losses, in cases (none) where the MLP-adapter does not exist. Across the three datasets, as the proposed loss functions were gradually incorporated, the model performance exhibited an increasing trend, with $\mathcal{L}_s$ and $\mathcal{L}_{\text{re}}$ demonstrating the most significant effects. However, for GC10-CLS, $\mathcal{L}_{b2c}$ resulted in performance degradation, possibly due to the absence of constraints on CLIP-V

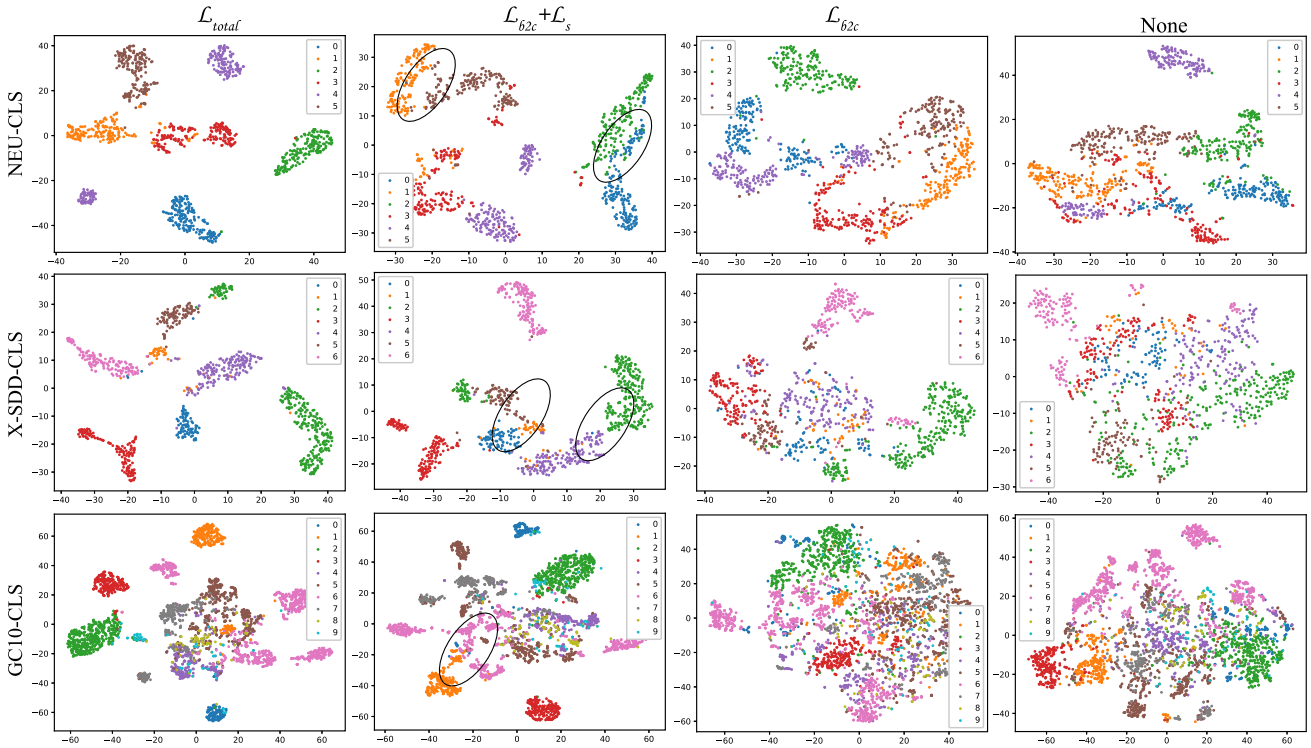
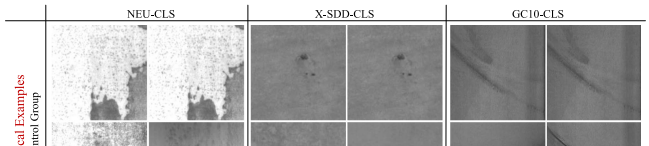


Fig. 6. Clustering visualization. The dots of the same color represent the same cluster, while the numbers in the legend indicate different clusters, as there are no true labels during the pretraining phase. The black ellipse represents the appearance of equivocal decision margins, which are alleviated by $\mathcal{L}_{\text{re}}$.

representation, leading to downstream models blindly learning more noisy data. Therefore, we proposed $\mathcal{L}_s$ to constrain the representation space. Furthermore, we delve into the behavior of data representation underlying the performance improvement of the model through qualitative analysis. Fig. 6 illustrates that transitioning from $\mathcal{L}_{b2c}$ to $\mathcal{L}_{b2c}$ + $\mathcal{L}_s$, the clustering effect of the model on unlabeled data becomes more compact, indicating that $\mathcal{L}_s$ has a significant compressive effect on the extensive CLIP-V representation space. The clustering effect of unlabeled samples in the second column of the Fig. 6 illustrates that, in the absence of $\mathcal{L}_{\text{re}}$, DM clusters exhibit partial overlap (as indicated by the black ellipse). This overlap arises from equivocal decision margin between clusters, making it challenging for samples with unclear distances within and between classes to escape this margin. Consequently, this overlap leads to a deterioration in defect recognition, as shown in Table V. Further transitioning to $\mathcal{L}_{\text{total}}$, there is less overlap in the representation of instances between different clusters, indicating that $\mathcal{L}_{re}$ has the ability to handle samples at the equivocal decision margin, effectively alleviating the problem of difficult discrimination of equivocal margin samples due to the ambiguity distance between intra-class and interclass. Importantly, CVGSSL effectively captures critical representations within the diverse CLIP-V representation space by leveraging three synergistic loss functions to guide DM. This guidance enables the DM's representation space to better encompass the shared characteristics of MSD data, leading to enhanced clustering performance. This improvement is evident in the transition from scattered clusters in the "none" stage to more cohesive clusters in the "$\mathcal{L}_{\text{total}}$" stage.

Furthermore, to gain a more intuitive understanding of the equivocal decision margin problem caused by the ambiguity of



Type	Different	Same	Different	Same	Different	Same
w/o $\mathcal{L}_{re}$	0.7500	0.7893	0.9105	0.8982	0.9247	0.9116
w/ $\mathcal{L}_{re}$	0.7253	0.7787	0.8490	0.8901	0.8818	0.8900

Fig. 7. Diagram illustrating the similarity between practical examples. The "different" column indicates that two images are similar but belong to different categories, while the "same" column denotes images that belong to the same category.

intraclass and interclass distances between instances, we discuss this by analyzing samples that appear similar but actually belong to different categories. In Fig. 7, "different" and "same" indicate that the control group belongs to different and same categories, respectively, with the numbers representing the similarity within the control group. As illustrated in Fig. 7, the similarity between "different" and "same" is not significantly different, causing DM to fall into an equivocal decision margin, making it challenging to distinguish between the two accurately. However, across all three datasets, there was a notable decrease in similarity after implementing $\mathcal{L}_{\text{re}}$, indicating that $\mathcal{L}_{\text{re}}$ effectively prevents instances from becoming overly similar. In particular, $\mathcal{L}_{\text{re}}$ exerts a stronger limiting effect on different yet similar samples, resulting in increased distance between "different" and

TABLE VI
ACCURACY IN ONE-SHOT TO FOUR-SHOT SETTINGS FOR DIFFERENT WEIGHTS OF LOSS FUNCTION

DATASETS	$[\mathcal{L}_{b2c}, \mathcal{L}_s, \mathcal{L}_{re}]$	One-shot	Two-shot	Three-shot	Four-shot
NEU-CLS	None	**92.21±5.00**	**96.05±2.43**	**97.09±0.86**	**97.41±0.86**
	[0.3,0.3,0.4]	83.98±7.82	90.87±4.05	92.83±2.29	93.87±1.79
	[0.3,0.5,0.2]	79.52±8.29	88.23±4.45	91.51±2.54	92.29±2.03
	[0.4,0.4,0.2]	83.41±7.20	89.95±3.96	92.82±2.59	93.81±1.77
	[0.5,0.3,0.2]	84.36±6.39	91.35±3.24	93.17±1.90	93.69±1.69
X-SDD-CLS	None	**81.70±7.89**	**89.39±4.16**	**91.23±3.00**	**92.27±2.23**
	[0.3,0.3,0.4]	77.08±6.16	83.83±5.39	86.27±4.06	88.64±2.53
	[0.3,0.5,0.2]	73.83±8.31	83.65±4.55	85.75±3.69	88.00±1.76
	[0.4,0.4,0.2]	72.49±8.62	83.42±6.18	86.87±2.64	88.18±2.08
	[0.5,0.3,0.2]	74.96±9.40	83.66±4.62	87.43±3.13	88.54±2.40

"same" after applying $\mathcal{L}_{re}$. This is attributed to $\mathcal{L}_{re}$ enabling DM to learn higher level and more abstract features, enhancing DM's understanding of common defect features, and effectively distinguishing samples with equivocal distances both within and between classes, thereby alleviating equivocal decision margins.

On the other hand, by assigning different weights to $\mathcal{L}_{b2c}$, $\mathcal{L}_s$, and $\mathcal{L}_{re}$, we discuss the emphasis of each loss function within the overall loss. In Table VI, "none" indicates no weight assignment. Since $\mathcal{L}_{b2c}$ and $\mathcal{L}_s$ function as a pair and are adversarial, we will initially set the weights of $\mathcal{L}_{b2c}$ and $\mathcal{L}_s$ to be equal and then analyze the performance impact when varying the weight of $\mathcal{L}_{re}$, either increasing or decreasing it. Specifically, when the weights are set to [0.3, 0.3, 0.4] and [0.4, 0.4, 0.2], only for the nonuniform dataset X-SDD-CLS does $\mathcal{L}_{re}$ receive a larger weight, resulting in better one-shot performance. Even when a weight configuration of [0.4, 0.4, 0.2] is used, with greater emphasis on $\mathcal{L}_{b2c}$ and $\mathcal{L}_s$ and a smaller weight for $\mathcal{L}_{re}$, it does not effectively reduce the risk of a equivocal decision margin. This suggests that when supervisory information is limited, $\mathcal{L}_{re}$ can better leverage DM's higher level representations to enhance the model's task relevance. Furthermore, weights from [0.3, 0.5, 0.2] and [0.5, 0.3, 0.2] indicate that for both NEU-CLS and X-SDD-CLS, a greater weight on $\mathcal{L}_{b2c}$ improves model performance. Intuitively, the significance of these three loss functions is challenging to optimize through manual intervention. In contrast, allowing the model to operate without interference in weight assignments can encourage it to achieve balance adaptively. However, assigning specific weights can artificially disrupt this balance. Consequently, the model often achieves optimal performance when no fixed weight is assigned.

D. Interesting Region Visualization

Visualizing the region of interest of the DM during the unlabeled pretraining stage provides insight into whether the DM has focused on relevant defect information. Fig. 8 demonstrates that for most defects, the DM accurately focuses on their shape and position, as evidenced by the second and last columns of defects in NEU-CLS. In addition, the DM appears to have acquired ViT's excellent boundary recognition and potential segmentation abilities, as seen in the third and third-to-last columns of defects in GC10-CLS. Overall, CVGSSL enables the DM to accurately focus on defect location, shape, and even boundary information, offering valuable insights for enhancing intelligent and systematic industrial quality inspection systems.

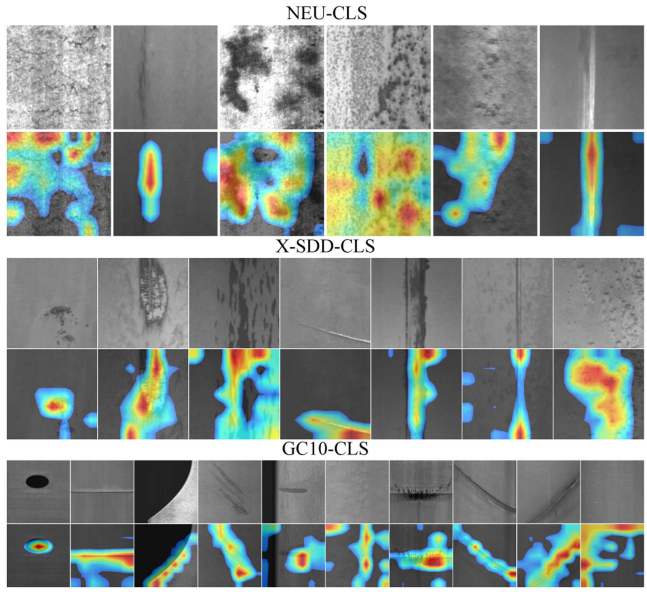


Fig. 8. Visualization of CVGSSL Interested region. The areas in red represent the regions that the model focuses on more prominently.

V. CONCLUSION

In this article, we introduced a novel self-supervised framework, named CVGSSL, tailored for few-shot MSDR. CVGSSL comprises two key parts. First, we utilized CLIP-V to provide rich and diverse characterization guidance for SSL. In addition, we constructed an MLP-adapter to adapt these characterizations. Second, we proposed a self-constrained loss to alleviate the negative impact of intraclass and interclass distance ambiguity on the DM learning process of MSD data. A large number of experiments demonstrate that CVGSSL is progressive and effective for unlabeled representation learning to few-shot MSDR. Specifically, the powerful representational ability of CVGSSL has not been explored across a broader spectrum of tasks, such as shape segmentation and position recognition. In addition, further theoretical analysis of model performance to achieve interpretable models is lacking. Therefore, in future work, we will endeavor to integrate the capabilities of CVGSSL into a broader range of tasks and conduct theoretical analyses to evaluate the model's generalizability.

REFERENCES

[1] H. Shang, J. Wu, C. Sun, J. Liu, X. Chen, and R. Yan, "Global prior transformer network in intelligent borescope inspection for surface damage detection of aeroengine blade," *IEEE Trans. Ind. Inform.*, vol. 19, no. 8, pp. 8865–8877, Aug. 2023.

[2] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Inform.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.

[3] D. Shan, Y. Zhang, S. Coleman, D. Kerr, S. Liu, and Z. Hu, "Unseen-material few-shot defect segmentation with optimal bilateral feature transport network," *IEEE Trans. Ind. Inform.*, vol. 19, no. 7, pp. 8072–8082, Jul. 2023.

[4] B. Yang, Z. Liu, G. Duan, and J. Tan, "Mask2Defect: A prior knowledge-based data augmentation method for metal surface defect inspection," *IEEE Trans. Ind. Inform.*, vol. 18, no. 10, pp. 6743–6755, Oct. 2022.

[5] K. Zhang, R. Cai, C. Zhou, and Y. Liu, "Debiased contrastive learning for time-series representation learning and fault detection," *IEEE Trans. Ind. Inform.*, vol. 20, no. 5, pp. 7641–7653, May 2024.

[6] G.-J. Qi and J. Luo, "Small data challenges in Big Data era: A survey of recent progress on unsupervised and semi-supervised methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2168–2187, Apr. 2022.

[7] Y. Liu, L. Bai, J. Ma, W. Wang, and W. Ouyang, "Self-supervised feature learning for appliance recognition in nonintrusive load monitoring," *IEEE Trans. Ind. Inform.*, vol. 20, no. 2, pp. 1698–1710, Feb. 2024.

[8] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[9] A. Y. Wang, K. Kay, T. Naselaris, M. J. Tarr, and L. Wehbe, "Better models of human high-level visual cortex emerge from natural language supervision with a large and diverse dataset," *Nature Mach. Intell.*, vol. 5, no. 12, pp. 1415–1426, 2023.

[10] Z. Wu et al., "Building an open-vocabulary video clip model with better architectures, optimization and data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 4747–4762, Jul. 2024.

[11] K. Vishniakov, Z. Shen, and Z. Liu, "ConvNet vs transformer, supervised vs CLIP: Beyond ImageNet accuracy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 49545–49557.

[12] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[13] Q. Luo, Y. Sun, P. Li, O. Simpson, L. Tian, and Y. He, "Generalized completed local binary patterns for time-efficient steel surface defect classification," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 667–679, Mar. 2019.

[14] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, *arXiv:2003.04297*.

[15] J. Lian et al., "Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network," *IEEE Trans. Ind. Inform.*, vol. 16, no. 2, pp. 1343–1351, Feb. 2020.

[16] L. Wen, Y. Wang, and X. Li, "A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8988–8998, Dec. 2022.

[17] H. Dong, K. Song, Q. Wang, Y. Yan, and P. Jiang, "Deep metric learning-based for multi-target few-shot pavement distress classification," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1801–1810, Mar. 2022.

[18] X. Sun, S. Yang, and C. Zhao, "Lightweight industrial image classifier based on federated few-shot learning," *IEEE Trans. Ind. Inform.*, vol. 19, no. 6, pp. 7367–7376, Jun. 2023.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[20] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15745–15753.

[21] Y. Xu et al., "Flexible and diverse contrastive learning for steel surface defect recognition with few labeled samples," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 2516014.

[22] T. Wang et al., "Few-shot steel surface defect recognition via self-supervised teacher–student model with min–max instances similarity," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 5026016.

[23] H. Wei et al., "Vary: Scaling up the vision vocabulary for large vision-language models," in *Proc. Eur. Conf. Comput. Vis.*, 2023, pp. 408–424.

[24] M. Wortsman et al., "Robust fine-tuning of zero-shot models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 7949–7961.

[25] G. Kwon and J. C. Ye, "One-shot adaptation of GAN in just one CLIP," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12179–12191, Oct. 2023.

[26] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[27] M. Ye, X. Zhang, P. C. Yuen, and S. F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6203–6212.

[28] J. B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21271–21284.

[29] K. Sohn et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 596–608.

[30] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 9912–9924.

[31] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11552–11563.

[32] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14451–14461.