DefectSynth: Few-Shot Defective Image Generation by Modeling Shape and Appearance

Dexu Zhao, Xukun Qin, Xinghui Dong, *Member, IEEE*

***Abstract*—Since the acquisition of a large amount of defect data is expensive and time-consuming, the limited availability of defective samples impairs the accuracy and generalizability of defect detection methods. Although defect generation approaches have been explored for data augmentation, they usually suffer from either a lack of realism or limited diversity. To address these challenges, we propose a two-stage controllable few-shot[1] defective image generation network, namely, DefectSynth[2], which models both the shape and appearance of defects. The first stage aims to generate continuous defect masks even with a few real masks. To this end, we propose a Hybrid Mask Interpolation (HMI) module, which performs interpolation in the image or latent space. The second stage is used to synthesize the defective appearance. We first fine-tune the pre-trained ControlNet, which is then used together with a pre-trained stable diffusion model to synthesize defective images. Given a mask generated in the first stage and a text prompt, they are integrated with a defect-free image to synthesize a high-fidelity defective image. To alleviate the issue of generation of indistinct defects with existing methods, we propose a Selective Attention Enhancement (SAE) mechanism that highlights the details of defects. We also design a Similarity-Based Feature Fusion (SFF) module to merge different local features, thereby further enriching the appearance diversity of defects. Using the defect data generated by DefectSynth, the classification accuracy on MVTec-AD has been improved from 44.03% to 66.70% compared with the baseline without synthetic data augmentation, while the F1-Score values computed on GDXray and DeepCrack for small-defect localization have been increased from 70.48% to 76.64% and from 70.08% to 83.27%, respectively. These performance gains should be due to the fact that our method is able to generate realistic and diverse defective images by modeling both the shape and appearance of defects.**

***Note to Practitioners*—DefectSynth is a turnkey, two-stage approach, specifically designed for practitioners who need abundant, realistic defective images but only have a limited number of real samples. The first stage aims to generate defective masks. Using the Hybrid Mask Interpolation (HMI) module that we design, DefectSynth generates a variety of continuous and natural defective masks based on a limited number of real masks, addressing the issue of insufficient variations of the shape and position of defects. The second stage focuses on defective appearance synthesis, in which a high-fidelity defective image is synthesized guided by the mask generated in the first stage and a text prompt. To this purpose, we have introduced two key techniques, i.e., a Selective Attention Enhancement (SAE) mechanism and a Similarity-Based Feature Fusion (SFF) module. The SAE mechanism can identify areas that are not clearly generated and allocate more attention to them, ensuring that subtle cracks, pores and slight discolorations become more noticeable. The SFF module extracts local texture examples from the current batch and seamlessly integrates them into the evolving defect areas, creating a more diverse appearance. DefectSynth empowers the training of more effective defect detection models by delivering a large and rich corpus of defective images. In particular, it can generate high-quality images that bridge the gap between the abundance of normal samples and the demand for diverse defective images. As a result, our DefectSynth is beneficial for defect detection systems in real-world industrial scenarios.**

***Index Terms*—Defective Image Generation, Defect Generation, Data Augmentation, Diffusion Model, Defect Detection.**

This study was supported by the National Natural Science Foundation of China (NSFC) (No. 42576200) (Corresponding author: Xinghui Dong).

D. Zhao, X. Qin and X. Dong are with the State Key Laboratory of Physical Oceanography and the Faculty of Information Science and Engineering, Ocean University of China, Qingdao, 266100 (e-mail: zhaodexu@stu.ouc.edu.cn, xkqin@stu.ouc.edu.cn, xinghui.dong@ouc.edu.cn).

[1]Following prior few-shot defect generation studies, such as DFMGAN [1], "Few-Shot" means that no more than 30 real defective images are used.

[2]The source code and models are available at https://github.com/INDTLab/DefectSynth

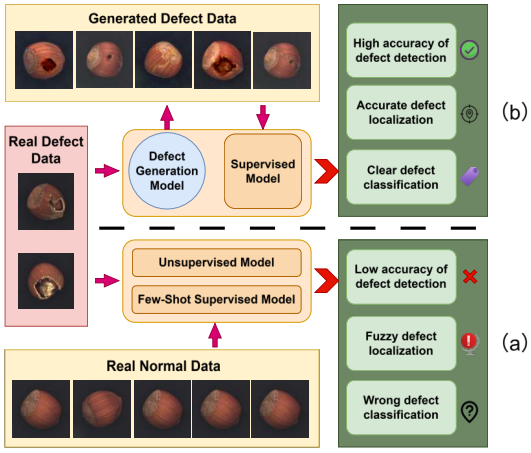

Fig. 1. Comparison between two different defect detection scenarios in which a small number of real defective images and/or normal images are used (a) and a large number of defective images that a defective image generation model synthesizes are utilized (b).

I. Introduction

In modern industrial production, defect detection is key to product quality and production efficiency [2]–[4]. Traditionally, defect detection methods were designed on top of hand-crafted features. However, these features usually lack robustness and the engineering of them is inefficient. In the past decade, deep learning techniques have shown significant promise in the field of defect detection [5]–[7]. With the continuous development of industrial automation and intelligence, the requirements for automated defect detection are increasing. An ideal defect detection system should be able to identify various types of defects during the production process in real-time and accurately [8]. Although a large number of normal samples can be obtained in real industrial production scenarios,

defective samples are rare. The severe class imbalance impairs the performance of the defect detection model (see Fig. 1(b)).

The scarcity of defective samples limits the learning ability of a model. In this situation, the model will struggle to capture the diversity and complexity of defects. This problem impairs the generalizability of the model and prevents it from adapting to varying defect types in real industrial production lines. In addition, the class imbalance issue will unavoidably cause the model to deviate from the characteristics of abnormal samples, further decreasing the detection accuracy.

To alleviate the scarcity of defective samples, researchers have proposed unsupervised [9], [10], semi-supervised [11], [12], and weakly supervised [13], [14] defect detection methods. However, these methods normally encounter limitations. Since unsupervised methods lack direct supervision from defective samples, it is challenging for them to locate and classify defects accurately. Due to the reliance on the specific features learned during the training process, the generalizability of semi-supervised defect detection methods is limited when confronted with unseen defect categories. Furthermore, the imprecision of coarse-grained labels used by weakly supervised methods prevents the model from learning sufficiently detailed defect features. A practical solution is to expand the defect dataset and enhance the performance of defect detection tasks through data augmentation.

Generally speaking, two categories of approaches can be used to increase the number of defective samples, including model-free generation and model-based generation. The model-free generation approach [4], [15], [16] normally involves randomly cropping and pasting existing defects onto a normal image. For example, the DRAEM method [15] pasted defect textures onto normal samples, while the CutPaste [16] method simulated defects by cutting and pasting local regions. These methods tended to produce similar patterns rather than samples with rich variations. The generated images often lack natural transitions with the background, resulting in noticeable inconsistencies, which probably have a negative impact on the performance of defect detection methods.

On the other hand, model-based generation methods were usually designed on top of deep learning techniques [1], [17], [18], which typically used Generative Adversarial Networks (GANs) [19], such as DefectGAN [17], ConGAN [18], and DFMGAN [1]. DefectGAN [17] generated defect samples by training a generator and discriminator. However, it required a large amount of training data. ConGAN [18] were unable to generate pixel-level masks and were only applicable to defect images of texture categories. DFMGAN [1] trained StyleGAN2 [20] on normal samples and used a small number of defective samples to train the mask and defect modules. Nevertheless, the alignment between the generated defects and masks is challenging. Recently, diffusion models have been applied to defect generation [3], [21]–[25]. For example, AnomalyDiffusion [21] introduced spatial anomaly embeddings, achieving better alignment between generated image-mask pairs. Due to unreasonable positions and shapes in the generated masks, unrealistic defect samples may be produced. In this case, it is crucial to build a defect generation network that produces realistic defective images and masks.

To address the challenge of generation of realistic and diverse defective images in a few-shot setting, we propose DefectSynth, a controllable two-stage framework. By decoupling the synthesis into defective mask generation and defective appearance synthesis, our approach effectively captures both geometric and textural nuances. Specifically, a Hybrid Mask Interpolation (HMI) module is introduced in the first stage to ensure structural continuity and diversity. In the second stage, we propose a Selective Attention Enhancement (SAE) mechanism and a Similarity-Based Feature Fusion (SFF) module. They address the issues of inconspicuous defect prominence and limited textural variety, respectively. Our DefectSynth enables the generation of high-quality and diverse defective images with limited real training samples.

The defective images that our method generates can be utilized as a data augmentation solution for defect detection tasks. To our knowledge, defective image generation has not been fulfilled by jointly exploiting both the shape and appearance cues. The main contributions of this study can be summarized as fourfold.

(1) We propose a controllable few-shot defective image generation network, i.e., DefectSynth. This network is designed based on a two-stage scheme. In contrast to existing methods, it utilizes both the shape and appearance cues of defects.

(2) We design a Hybrid Mask Interpolation (HMI) module which performs the interpolation operation in either the image or latent space. Since this module generates a set of vivid intermediate masks through smooth image interpolation, it provides more information of the shape and location of defects.

(3) We introduce a Selective Attention Enhancement (SAE) mechanism, which aims to allocate more attention to the regions where the defects synthesized are subtle. As a result, it addresses the problem of inconspicuous defect generation.

(4) We adopt a Similarity-Based Feature Fusion (SFF) module, which randomly selects a base image and a reference image within a mini-batch and replaces the matched local features in the reference image by those in the base image. This module can combine the local features of different images to generate defective images with greater diversity and realism.

The rest of this paper is organized as follows. Related work is reviewed in Section II. The proposed DefectSynth is introduced in Section III. The defective image generation and downstream task experiments are reported in Sections IV and V, respectively. Finally, we draw our conclusion in Section V.

II. Related Work

A. Image Generation

Early Variational Autoencoders (VAEs) [26] and GANs [19] have achieved significant success in the field of image generation. VAEs encode input images into distribution parameters in the latent space via an encoder and reconstruct images through a decoder, thereby continuously optimizing the quality and diversity of the generated images. Many improved VAE variants were proposed to enhance the generative capability [27], [28]. However, they generally suffered from limitations, such as insufficient image quality and details, limited diversity, and complex and unstable training.

GANs generate realistic images through adversarial training between generators and discriminators. For instance, Deep Convolutional GANs (DCGANs) [29] improved the quality of generated images by introducing convolutional layers. By incorporating a stylized network architecture, StyleGANs [30] generated detailed and diverse images, such as realistic human faces. BigGANs [31] generated high-quality natural images, covering various categories from animals to landscapes, through large-scale training and improved architectures. Although GANs have made great progress in image generation, they generally encountered challenges, such as unstable training, pattern collapse, and overfitting to the limited data.

Recently, diffusion models, an emerging generative model with a Markov chain structure, have been widely applied to image generation and multimodal generation tasks. Compared to GANs, these models do not require complex adversarial training, making them easier and more stable to train [32]. Additionally, samples generated by those models typically exhibit higher quality and diversity, better capturing the details and structure of the data. Furthermore, Latent Diffusion Models (LDMs) [33], built based on Denoising Diffusion Probabilistic Models (DDPMs) [34], accelerated the generation process by introducing a low-dimensional latent space and combining it with autoencoders, achieving efficient text-to-image synthesis. DALL·E [35] generated high-quality images based on textual descriptions by combining diffusion models with Transformer [36]. Stable Diffusion [33] improved this method using more efficient architectures and optimizations [33]. In [37], ControlNet introduced conditional control mechanisms, enabling the images generated meet user-specified conditions. DiffusionCLIP [38] leveraged the text understanding ability of the Contrastive Language-Image Pre-training (CLIP) [39] model, improving the quality and relevance of text-to-image generation.

B. Defective Image Generation

Acquisition of a large number of high-quality defect images is typically costly and challenging. To address this issue, defective image generation techniques were developed. Traditional methods [4], [15], [16], [40] usually employed CutPaste [16] and crop-and-paste approaches to generate new samples. However, these methods produced overly simplistic defects that could not cover all possible defect scenarios. Deep learning methods have been utilized in defective image generation. As early attempts, Variational Autoencoders (VAEs) were used to generate defect data [41], [42]. Yun et al. [41] proposed a Conditional Convolutional Variational Autoencoder (CCVAE) to address the shortage of defective data in metal manufacturing. Fu et al. [43] proposed SAVAE, a self-attention adversarial VAE for rail surface defect expansion, which was tailored to rail scenarios with limited generalization to other defect types.

GANs were also employed to generate realistic defective images through adversarial training. For example, FCGAN [44] used an attention mechanism to distinguish between the foreground and background and synthesized pseudo-defective images. SDGAN [45] and Defect-GAN [17] generated defects in standard samples by learning from the defect data, but they required large amounts of defect data and could not produce defect masks. DFMGAN [1] achieved few-shot defect image generation by attaching defect-aware residual blocks to a pre-trained StyleGAN2 [20]. However, it suffered from issues, such as insufficient realism, inaccurate alignment between anomalies and masks, and limitations in the size of generated images. In [46], a DCGAN-derived framework was proposed for apparel stitching defect synthesis, but it lacked flexibility in adapting to diverse defect shapes due to reliance on traditional adversarial training. Jiang et al. [47] introduced MGDefect, a mask-guided dual-module method, generating pixel-level annotated defects with limited samples, while increasing model complexity via dual discrimination.

On the other hand, diffusion models have shown great potential in generation of defective images [21]–[25]. In [21], AnomalyDiffusion decomposed abnormal information into the appearance and location data through spatial anomaly embedding. An adaptive attention re-weighting mechanism was used to generate high-quality and diverse abnormal images. Yu et al. [22] merged the latent space of defect-free and defective samples and combined a response alignment strategy, a defect moving strategy, and a regional average loss for defect generation. In [23], a semantic-rich defect spectrum framework was adopted for large-scale data sets. Shi et al. [24] proposed a text-guided diffusion method, i.e., DefectDiffu, to model both intra-product background consistency and inter-product defect consistency and modulate the consistency perturbation directions. Dai et al. [25] introduced a unified few-shot model with separation-sharing fine-tuning, i.e., SeaS, generating diverse anomalies, consistent normals, and precise masks.

However, the above-mentioned methods still suffered from issues, such as boundary overstepping, shape collapse, and lack of structural integrity. In contrast, the proposed DefectSynth achieves better diversity and realism through a two-stage generation process.

III. DefectSynth

Existing defective image generation methods often struggle to achieve realism and diversity. To address these limitations, we propose a two-stage few-shot defective image generation network, referred to as DefectSynth, which models both the shape and appearance of defects. The first stage aims to generate diverse, natural defect masks. We introduce a Hybrid Mask Interpolation (HMI) module, to ensure smooth transitions and enhance the diversity of the location and shape of defects. The second stage is used to fulfill defective appearance synthesis using the Stable Diffusion model [33]. ControlNet [37] is also incorporated into this stage, which guides the generation of defective images. In addition, we design a Selective Attention Enhancement (SAE) mechanism and a Similarity-Based Feature Fusion (SFF) module, to capture local characteristics and increase the diversity of defect appearances. The architecture of DefectSynth is shown in Fig. 2.

A. Preliminaries

The Diffusion Model (DM) is a generative method that produces high-quality samples by reversing a noise diffusion

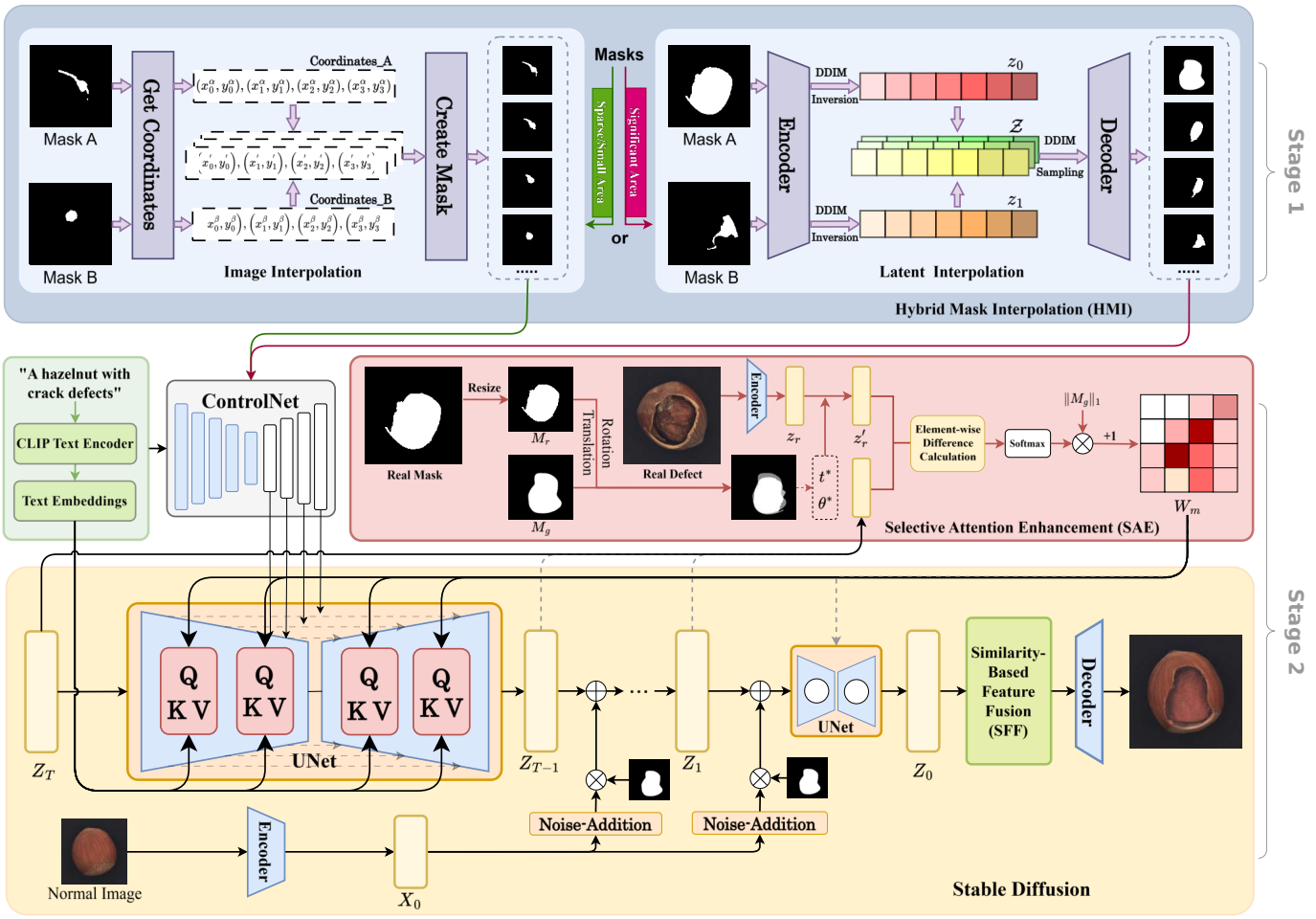

Fig. 2. The architecture of the proposed DefectSynth, which contains a defect mask generation stage (i.e., stage 1) and a defective image synthesis stage (i.e., stage 2). Within stage 1, diverse and spatially vivid intermediate masks are generated via image or latent interpolation. Within stage 2, a normal image, the mask generated in the first stage and a text prompt are received. A CLIP [39] text encoder converts the prompt to a set of textual embeddings. The masks and embeddings are sent to a ControlNet-guided UNet denoising network. The result is a defective image that exhibits clear anomalies while blending naturally with the normal background.

process. The method transforms noise into data samples using the reverse diffusion steps learned. In the DM [34], a sample $x_0 \sim p_{\text{data}}(x)$ undergoes T noise-addition steps:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \quad t = 1, ..., T, \quad (1)$$

where β_t controls noise magnitude at step t, increasing with t until x_T approaches $\mathcal{N}(0, \mathbf{I})$.

The model learns a reverse process parameterized by a neural network. This operation can be expressed as

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (2)$$

The training process minimizes the variational lower bound:

$$\mathbb{E}_q \left[\sum_{t=1}^{T} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \right] \quad (3)$$

where $\mathbb{D}_{KL}$ measures distribution divergence. This optimization operation enables the model to generate samples matching the true data distribution.

B. Overall Network Architecture

As shown in Fig. 2, our DefectSynth consists of a defect mask generation stage (stage 1) and a defective appearance synthesis stage (stage 2). In stage 1, a Hybrid Mask Interpolation (HMI) module is proposed, which takes two masks as input and performs an interpolation operation in either the image or latent space. This dual-space design ensures smooth transitions and generates continuous intermediate masks, which capture rich variations of the shape and position of defects. As a result, the diversity of the shape and position of defects is expanded, providing rich and diverse structural priors for the subsequent stage.

Controllable defective appearance synthesis is performed in stage 2. We first fine-tune the pre-trained ControlNet [37] using the mask generated in stage 1, the normal image y, the defect mask m, the control text prompt c and the real defective image along with the corresponding mask. Specifically, the text prompt is first converted into a set of semantic embeddings using the text encoder of CLIP [39], providing the high-level semantic guidance for the generation process. The mask generated in stage 1 also serves as a spatial condition input to

ControlNet, which fed additional control signals into the UNet to achieve precise control at the pixel level over the shape and position of defects.

Then the pre-trained Stable Diffusion model [33] is used to generate realistic and diverse defective images. The ControlNet fine-tuned is used to ensure precise control over the position, shape and category of the defects to be synthesized, with the guidance of the mask generated in stage 1 and a text prompt. For the sake of improving the quality and diversity of the defective images generated, a Selective Attention Enhancement (SAE) mechanism and a Similarity-Based Feature Fusion (SFF) module are adopted. The SAE mechanism calculates the differences between real and generated defects in the latent space, which dynamically directs more attention to under-generated regions, thereby enhancing the structural accuracy and visual prominence of subtle defects. In contrast, the SFF module enriches textural diversity by performing local feature replacements based on the similarity between different defects.

During the training phase, we fine-tune the ControlNet, which is attached to the pre-trained and frozen Stable Diffusion v1.5 UNet. The training objective is to minimize the standard denoising loss. As a result, the model learns to reconstruct the clean defective image from a noisy latent representation, conditioned on the given mask and text prompt. In the inference phase, the frozen Stable Diffusion model, guided by the fine-tuned ControlNet, synthesizes a defective image through deterministic DDIM sampling, given a normal image, a mask generated in stage 1 and a text prompt.

C. Hybrid Mask Interpolation Module

Two defect masks with different positions and shapes are selected from the real masks of the same defect category. These masks serve as input for mask generation. The defects contained in them can be cracks, notches, corrosion and other categories of abnormalities. In terms of the two masks, we first obtain their latent representations using a Variational Autoencoder (VAE) [26], followed by the Denoising Diffusion Implicit Model (DDIM) inversion [48] to acquire the latent noises of them, i.e., z_0 and z_1, respectively. Then an interpolation operation is performed between them using spherical linear interpolation (slerp) [49], to derive a set of intermediate latent noises $\mathcal{Z}$. This process can be expressed as:

$$\mathcal{Z} = \{z_\alpha \mid z_\alpha = \mathrm{Slerp}(z_0, z_1, \alpha), \alpha \in [0,1]\}, \qquad (4)$$

where $\alpha \in [0,1]$ is the interpolation coefficient used to control the degree of transition. The slerp operation can maintain a smooth trajectory in high-dimensional space, preserving the geometric structure of the latent manifold [50]. After the latent noise has been generated, we perform a reverse diffusion process using a conditional diffusion model, which restores the image sequence by gradually removing noise.

However, latent noise interpolation usually produces all-black intermediate images in the occurrence of small white regions. To address this issue, we propose an image-space interpolation method for defect mask generation. This approach operates directly on the pixel coordinates of binary masks, which ensures smooth transitions between different defect patterns. The image-space interpolation begins by extracting white pixel coordinates from both the source and target masks. We normalize the coordinate sets to the equal length by linearly resampling the shorter coordinate sequence to the same number of points. Then intermediate positions are computed using weighted averaging between corresponding coordinate pairs, where the weighting factor α controls the progression from the source value (α = 0) to the target value (α = 1).

To maintain spatial coherence during transitions, we introduce a continuity optimization that analyzes the spacing between adjacent interpolated points. When the distance between consecutive points exceeds a small threshold, additional intermediate points are inserted through uniform sampling along the connecting line segment. This process avoids discontinuous jumps and ensures smooth morphological evolution. The interpolated coordinate set is converted into a binary image. As a result, a series of intermediate transition masks can be generated using image-space interpolation.

The HMI module can perform interpolation in either the image space or the latent space. To determine which space is selected, we compute the average area ratio of defective regions relative to the image size for each defect category. Specifically, let $\bar{A}_{\text{mask}}$ denote this average area ratio. If $\bar{A}_{\text{mask}} \geq \tau$ (that we empirically set $\tau = 0.053$, which indicates that the defects in the given category (e.g., stains and corrosion) are relatively large, latent space interpolation will be used to enable smooth semantic transitions. Otherwise, i.e., $\bar{A}_{\text{mask}} < \tau$, which corresponds to defect categories with relatively small areas (e.g., fine cracks and tiny holes), image space interpolation will be used to preserve structural continuity. The selection strategy balances shape authenticity and transition smoothness, outperforms the random pattern generation method, such as Perlin noise [15], [51]. The HMI module ensures the generation of natural and plausible masks across various defect categories. When applied to defective image generation tasks, these masks can guide the image synthesis process through a progressive defect morphology transformation.

D. Selective Attention Enhancement Mechanism

To better match the defective areas of the generated image with the real image and guide the generation model to focus on areas where the defects are not obvious, we introduce a Selective Attention Enhancement (SAE) mechanism. Unlike existing attention re-weighting approaches (e.g., AnomalyDiffusion [21]) that aim to ensure spatial coverage of the provided mask, the SAE mechanism addresses a distinct challenge, i.e., visually subtle defects often lack prominence due to the denoising bias of diffusion models. Therefore, the SAE mechanism is designed for visual enhancement rather than spatial filling. This mechanism pays more attention to the generated regions that show a larger difference in appearance from the real defects during the denoising process.

To be specific, a real defective image and its corresponding binary mask are randomly selected from the training set and mapped into the latent space to obtain z_r and M_r. Given the current denoising time step, we have the latent representation z_t and the corresponding binary mask M_g. We align the

defective regions of z_r and z_t through the optimal translation and rotation operations as much as possible, to calculate the element-wise differences between the two regions. The optimal translation vector t^* and the optimal rotation angle θ^* are obtained as follows:

$$t^*, \theta^* = \arg\max_{t,\theta} \sum_{i,j} M_{r_{t,\theta}}(i,j) \cdot M_g(i,j) \tag{5}$$

where $M_{r_{t,\theta}}$ represents the transformation of M_r by the translation vector t and the rotation angle θ, and $M_g(i,j)$ is the value of M_g at the element position (i,j). This optimization aims to find a set of optimal spatial transformation parameters t^* and θ^* in order that the transformed $M_{r_{t,\theta}}$ has the maximum overlap with M_g in the defective regions.

The optimal spatial transformation parameters are applied to z_r to obtain the aligned latent representation z_r':

$$z_r' = R_{\theta^*}(T_{t^*}(z_r)) \tag{6}$$

Through this operation, we can align the real defective latent representation z_r to the current latent representation z_t in the defective regions. The weight map w_m is calculated based on the element-wise differences between the aligned z_r' and z_t within M_g as follows:

$$w_m = \|M_g\|_1 \cdot \text{Softmax}\Big(\text{mean}\Big((M_g \odot (z_r' - z_t))^2, \\ \dim = 1\Big)\Big) + 1 \tag{7}$$

For the regions within the mask that have a larger difference from the real defects, the generated defects in these regions are less obvious. According to Equation (7), higher weights are assigned to these regions. During the generation process, the weight map w_m is applied to the cross-attention module to dynamically adjust the attention weights, ensuring that the model focuses more on the high-weight regions of the weight map. Given the query vector Q, the key vector K, and the value vector V, the re-weighted cross-attention computation process can be expressed as:

$$\text{RW-Atten}(Q,K,V) = w_m \odot \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \tag{8}$$

In this way, the model dynamically assigns higher attention weights to the regions with less obvious defects during the generation process, effectively improving the structural integrity of the defects generated.

Algorithm 1 illustrates the computational process of the SAE mechanism. This mechanism performs spatial alignment (Equations (5)-(6)) once per generated image, obtaining an aligned real latent reference z_r'. At each denoising step, the weight map w_m is recomputed via Equation (7) using the current latent z_t, and is then used to multiplicatively modulate the cross-attention scores via element-wise multiplication (Equation (8)). This design ensures adaptive enhancement of subtle defects throughout the generation process. In essence, the SAE mechanism complements the spatial guidance provided by ControlNet [37]. While ControlNet ensures that the generated defect conforms to the input mask in terms of overall structure and location, SAE focuses on enhancing the visual prominence of subtle defective regions (e.g., fine cracks and faint discolorations) that may otherwise be under-generated due to the smoothing tendencies of diffusion models.

Algorithm 1 Integration of SAE into Denoising Loop (/Image)

Require: Real latent z_r, real mask M_r, initial generated mask M_g
Ensure: Denoised image with enhanced defect prominence
Step 1: One-Time Alignment (before Denoising Loop)
Compute optimal translation t^* and rotation θ^* via Eq. (5)
$z_r' \leftarrow R_{\theta^*}\big(T_{t^*}(z_r)\big)$ //Align real latent to M_g
Step 2: Denoising Loop with SAE (for $t = T, \ldots, 1$**)**
for Each denoising step t **do**
  Obtain current noisy latent z_t
  SAE Weight Map Generation:
  $\Delta z \leftarrow M_g \odot (z_r' - z_t)$ // Latent difference within mask
  Compute $w_m^{(t)}$ via Eq. (7) using Δz
  SAE-Modulated Attention:
  $A \leftarrow \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)$
  $A' \leftarrow w_m^{(t)} \odot A$ //Multiplicative modulation
  Output $\leftarrow A' \cdot V$
  Continue remaining UNet operations and update z_{t-1}
end for
return Final denoised image

E. Similarity-Based Feature Fusion Module

To create more diverse and structurally plausible appearances in the generated defective regions, we propose a Similarity-Based Feature Fusion Module (SFF). This module operates on the latent representation $z_0 \in \mathbb{R}^{B\times C\times H\times W}$ after the denoising process is complete, enhancing the diversity of local features through iterative feature fusion within the current batch. For simplicity, we illustrate the workflow of the SFF module using a single sample. Given that the i-th sample b_i is selected from the latent representation z_0 in the current batch as the base feature $f \in \mathbb{R}^{C\times H\times W}$, the remaining $B-1$ samples are defined as the reference feature set $R = \{r_0, r_1, ..., r_{B-2}\}$. The module performs the following operations in turn.

Random Sampling of Feature Points: According to a preset sampling rate, $\text{np} = \text{rate} \times H \times W$ feature points are randomly sampled from f to form the base set of feature points $f_{\text{select}} \in \mathbb{R}^{C\times \text{np}}$.

Similarity Computation: For each reference feature $r_j \in R$, a similarity matrix between f_{select} and all spatial feature points in r_j is computed using the dot product operation.

Feature Matching: For each feature point in f_{select}, the most similar feature point is selected within each r_j. All the matching points corresponding to the base feature points in r_j are grouped into a best-matched point set $r_{\text{select}}^{(j)} \in \mathbb{R}^{C\times \text{np}}$.

Weighted Fusion and Replacement: The base feature points and all matched reference feature points are fused using two weights:

$$f_{\text{fused}} = w_i \cdot f_{\text{select}} + \sum_{j=0, j\neq i}^{B-1} w_j \cdot r_{\text{select}}^{(j)}, \tag{9}$$

where w_k represents preset non-negative weights, the index k corresponds to each sample in the batch (whether it serves as the base sample i or a reference sample j), and $\sum_{k=0}^{B-1} w_k = 1$. Finally, f_{fused} is replaced back into the corresponding sampling positions of the base feature f, completing the feature fusion for the sample b_i.

To achieve collaborative enhancement of all samples within the batch, the SFF module adopts an iterative fusion strategy by applying the above process to each sample in the batch:

- **For** $i = 0$ **to** $B - 1$ **do:**
  1) Treat the current sample b_i as the base feature f, and the remaining samples as the reference feature set R;
  2) Execute the aforementioned four operations to update b_i.
- **End For**

After the iteration has been complete, the original batch z_0 is transformed into a new fused batch z_0', where each sample has incorporated similar features from the other samples within the same batch. Consequently, the texture diversity of local regions is effectively enhanced while the semantic coherence of them is preserved.

IV. Defective Image Generation Experiments

To comprehensively evaluate the performance and generalizability of the proposed method, we conducted a series of experiments on defective image generation using the MVTec-AD [52], GDXray [53] and DeepCrack [54] data sets. These data sets cover various defect categories, imaging techniques and application scenarios. We will introduce the experimental setup, experimental results and ablation study as follows.

A. Experimental Setup

1) Data Sets: Following the existing studies [1], [21], [22], we used the MVTec-AD [52] data set. To augment our experiments, we also utilized the GDXray [53] and DeepCrack [54] data sets. These data sets span multiple industrial imaging modalities, diverse defect morphologies and scales, and varying annotation granularity and task difficulty levels.

MVTec-AD: As a high-quality data set widely adopted for defect detection and image generation, MVTec-AD [52] covers 15 everyday objects and textures, e.g., screw, wood, cable, carpet and leather. This data set contains 73 defect categories, ranging from subtle scratches and dents to structural deformations and missing parts. Each image in the MVTec-AD data set was annotated with a mask. Due to rich variations in object rigidity and texture regularity, this data set is ideal for evaluating a model across a wide spectrum of complexities.

GDXray: GDXray [53] is a public X-ray image data set designed for non-destructive testing (NDT). It contains 19,407 images divided into five categories, including castings, welds, baggages, nature and settings. The X-ray images of welds typically have low contrast and the defects contained are small. As a result, localization of defects is particularly challenging. The use of this modality allows us to explicitly test the robustness of a model under extreme imaging conditions and fine-grained anomalies. Since Stable Diffusion 1.5 [33] normally performed the best when generating square images, we cropped sub-images from the weld category in order to construct a new subset, which included defective weld images, associated masks and defect-free weld images. All of these images were resized to a resolution of 512×512 pixels and organized in the same file structure as that of MVTec-AD [52].

DeepCrack: The images of large-scale road and building surface cracks were included in DeepCrack [54]. We selected 21 cracked images and synthesized paired defect-free counterparts for data augmentation. These images provided complementary challenges to the compact, blob-like defects contained in the MVTec-AD [52] and GDXray [53] data sets.

2) Baselines: We employed six baseline methods for comparison across the MVTec AD [52], GDXray [53], and DeepCrack [54] data sets, including Crop Paste [40], DFMGAN [1], AnomalyDiffusion [21], Defect-Gen [23], DefectDiffu [24], and SeaS [25].

3) Evaluation Metrics: We employed three evaluation metrics on the MVTec-AD, GDXray and DeepCrack data sets, including Kernel Inception Distance (KID) [55], Learned Perceptual Image Patch Similarity (LPIPS) [56] and Inception Score (IS) [57]. Specifically, KID measures the distribution similarity between the generated and real images; LPIPS is a perceptual similarity metric, which evaluates image quality by comparing the distances between these images in the feature space; and IS provides a complementary image-level assessment for overall clarity and diversity.

We generated 1,000 defective images for each defect category. Each metric was computed between these images and the real defective images contained in the associated data set three times, and the average was reported.

4) Implementation Details: In this study, "few-shot" means that no more than 30 real defective images per category were used across all stages of our method. In the first stage of DefectSynth, we randomly selected two real defective masks from each category of the data set, except for the transistor category in the MVTec-AD data set [1] in which two distinct masks were deliberately specified because the variations in the shape and position of the defects were small. Mask interpolation was used to produce 10 intermediate masks per pair, yielding roughly 100 interpolations and approximately 1,000 masks in total. During the second stage, we used all available real defective images per category, which range from 8 to 30 images depending on the size of a category in the data set, together with their corresponding text prompts, to fine-tune the ControlNet [37] integrated with Stable Diffusion [33]. The number of defective samples per category used during the first and second stages is shown in Table I. All baselines are evaluated under the same data setting for a fair comparison.

Each image was resized to 512×512 pixels. Both the masks generated in the first stage and the text prompts were used as guidance. The mini-batch size was set to 2. We set the maximum number of epochs used for training to 2,000. The learning rate was fixed at 10^{-5}. A cosine noise schedule was employed with $T = 1{,}000$ forward diffusion steps. Within the inference phase, deterministic DDIM sampling [48] was adopted with $K = 20$ denoising steps and a

TABLE I
THE NUMBER OF REAL DEFECTIVE SAMPLES USED IN THE TWO STAGES.

Category	Defective Masks (Stage 1)	Defective Images (Stage 2)
Bottle	20 – 22	20 – 22
Cable	10 – 14	10 – 14
Capsule	20 – 23	20 – 23
Carpet	17 – 19	17 – 19
Grid	11 – 12	11 – 12
Hazelnut	17 – 18	17 – 18
Leather	17 – 19	17 – 19
Metal Nut	22 – 25	22 – 25
Pill	9 – 26	9 – 26
Screw	23 – 25	23 – 25
Tile	15 – 18	15 – 18
Toothbrush	30	30
Transistor	10	10
Wood	8 – 21	8 – 21
Zipper	16 – 19	16 – 19

Here, a single number indicates that the category contains only one defect type or all defect types in the category have the same number of samples.

randomly generated seed. Both networks were implemented using PyTorch 1.12.1. All experiments were conducted on an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory.

B. Experimental Results

1) Quantitative Evaluation: The KID$\times 10^3$, LPIPS and IS values obtained using six baselines and our method on the MVTec-AD [52] data set are shown in Table II. In terms of the KID$\times 10^3$ metric, our method achieved the best average performance. It is indicated that the images generated using our method are the closest in distribution to the real defective images, demonstrating excellent image similarity. Considering the LPIPS metric, the average performance of our method ranked the third and was only surpassed by AnomalyDiffusion and SeaS. This result demonstrates that the images generated by our method exhibit high visual divergence from the real defective images, thereby showcasing good diversity. Regarding the IS metric, our method ranked first together with SeaS. It is indicated that the images generated by our method performed prominently in overall clarity, category consistency and sample diversity, and it generated defective images with better quality and diversity. This finding further verifies the effectiveness of our method in image generation tasks.

When the GDXray [53] and DeepCrack [54] data sets were used, the KID$\times 10^3$, LPIPS and IS values derived using six baselines and our method are reported in Table III. As can be seen, our method always achieved the best performance in terms of the three metrics.

It should be noted that the KID value produced by the Crop-Paste [40] method was ignored because the images generated using this method were almost identical in appearance and the KID value computed in terms of these images was nearly zero. Generally speaking, our method normally outperformed, or at least performed comparably to, its counterparts across the three data sets no matter which metric was considered. The results demonstrate the strong capability of our method in rendering high-quality and diverse defective images.

2) Qualitative Evaluation: To intuitively display the results produced by the proposed method and six baselines for MVTec-AD [52] defective image generation, we present the images generated in Fig. 3. As a representative non-generative method, Crop-Paste generated images by cropping and pasting, which produced images with poor coherence and naturalness. Although DFMGAN generated better image quality without overfitting or mode collapse, it produced unreasonable defects on the grid glue data and had poor integration between defects and the background (see the result associated with the wood category). AnomalyDiffusion generated defective images with better diversity. However, it tended to produce illogical defects (as shown with regard to the carpet category) and struggled to generate small defects (such as grid glue and tile cracks) in which only the background was produced. Defect-Gen [23] tended to generate unnatural defects, often exhibiting distortions in both background and defect regions, such as texture deformations and irregular shapes in the cable and tile categories. DefectDiffu [24] produced defects that lack sufficient saliency, making them be easily overlooked or blended into the background, particularly for small-scale defects, e.g., glue on grid surfaces. In addition, SeaS [25] occasionally suffered from misalignment between the generated mask and the defect region (e.g., in the wood category), which compromised the visual coherence of the generated images and the usefulness for downstream tasks. In contrast, our method generated more realistic and well-shaped defective images and was able to generate small defects.

Fig. 4 further shows the images generated using the six baselines and our method in terms of the GDXray [53] and DeepCrack [54] data sets. Considering the GDXray generation task, it can be observed that the images generated by Crop-Paste exhibit prominent boundary artifacts between defects and the background. The images produced by DFMGAN lack coordination with the background and contain noise in normal regions, leading to poor naturalness. AnomalyDiffusion generated less distinct images with regard to fine defects, often producing blurry defective regions that failed to reveal the structure of weld defects. Defect-Gen generated weld defects that deviated from the background of real images, and the resulting defect regions often did not blend naturally with the surrounding environment. DefectDiffu produced defects that were insufficiently prominent, with low contrast against the background, failing to capture the distinct structural characteristics of the weld defects. Due to mask-defect misalignment, the defect regions generated by SeaS did not spatially correspond to the indicated mask positions, which reduced the usability of the generated defective samples. Compared with these baselines, our method generated images with clear defective characteristics. The defects generated were not only naturally-shaped but also well blended into the background.

Regarding the DeepCrack generation task, our method generated crack images with natural shapes that were well-blended into the road background. As a result, the interference from the background was avoided. In contrast, Crop-Paste generated images with boundary artifacts while DFMGAN produced images with color discrepancies. Incomplete cracks were generated by AnomalyDiffusion. Defect-Gen failed to

TABLE II
COMPARISON BETWEEN SIX BASELINES AND OUR METHOD IN THE DEFECT IMAGE GENERATION PERFORMANCE IN TERMS OF THE KID$\times 10^3$, LPIPS AND IS VALUES OBTAINED ON THE MVTEC-AD [52] DATA SET.

Category	**Crop-Paste** [40]			**DFMGAN** [1]			**AnomalyDiffusion** [21]			**Defect-Gen** [23]			**DefectDiffu** [24]			**SeaS** [25]			**Ours**		
	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑
Bottle	-	0.04	1.40	**70.90**	0.12	1.57	146.55	0.17	1.41	135.96	0.09	1.33	94.02	0.13	1.47	120.47	**0.25**	**1.76**	103.44	0.12	1.37
Cable	-	0.25	1.72	53.96	0.25	1.87	152.43	**0.40**	2.02	449.76	0.35	1.78	62.29	0.32	1.82	**29.88**	0.34	2.13	38.84	0.39	**2.17**
Capsule	-	0.05	1.21	40.63	0.10	1.51	76.96	0.18	1.46	237.90	0.06	1.36	25.27	0.14	1.43	43.13	**0.23**	**1.53**	**20.32**	0.18	1.38
Carpet	-	0.11	1.03	**25.14**	0.13	1.06	155.67	0.22	1.09	99.12	0.14	1.06	45.75	0.22	1.07	46.81	0.19	1.10	43.44	**0.24**	**1.12**
Grid	-	0.12	1.74	101.04	0.13	1.93	67.36	**0.45**	2.16	77.77	0.24	1.63	57.47	0.42	**2.41**	**36.65**	0.41	2.20	40.30	0.36	2.21
Hazelnut	-	0.21	1.87	21.16	0.24	1.83	48.49	0.31	**2.34**	121.22	0.25	1.86	**19.01**	0.30	1.93	85.07	**0.35**	2.07	42.73	0.29	1.93
Leather	-	0.14	1.63	**75.85**	0.17	1.75	175.88	**0.40**	1.68	194.49	0.11	1.61	108.06	0.25	1.74	103.25	**0.40**	**2.06**	105.86	0.33	1.80
Metal Nut	-	0.15	1.49	**44.04**	0.32	**1.83**	178.85	0.27	1.57	385.63	**0.48**	1.48	61.54	0.23	1.51	67.75	0.28	1.61	51.74	0.26	1.64
Pill	-	0.11	1.39	123.72	0.16	1.45	74.37	0.23	**1.56**	374.70	**0.39**	1.38	69.30	0.19	1.51	65.19	0.21	1.40	**50.93**	0.23	1.53
Screw	-	0.16	1.09	9.53	0.13	1.51	26.31	0.30	1.97	182.11	**0.46**	1.73	**5.79**	0.32	2.03	43.40	0.29	1.76	42.09	0.31	**2.22**
Tile	-	0.20	1.85	85.28	0.23	1.81	318.24	**0.48**	1.91	**56.64**	0.38	1.83	89.52	0.46	1.94	161.06	0.40	1.88	86.19	0.46	**2.37**
Toothbrush	-	0.08	1.47	46.49	0.18	1.73	66.51	0.18	1.64	86.83	0.11	1.31	28.16	**0.22**	**1.79**	62.80	0.15	1.61	**9.67**	0.19	1.42
Transistor	-	0.15	1.40	88.31	0.25	1.58	130.84	0.30	1.51	249.14	0.13	1.34	73.86	0.21	1.51	**64.11**	**0.36**	1.54	65.63	0.32	**1.68**
Wood	-	0.23	1.87	68.12	0.34	1.98	142.83	0.35	1.95	261.85	0.12	1.89	**43.23**	0.34	2.01	73.50	**0.41**	**2.11**	53.40	0.36	2.03
Zipper	-	0.11	1.21	77.67	0.27	1.38	132.36	0.24	1.33	115.41	0.25	1.27	**66.49**	0.22	1.32	109.87	**0.29**	**1.51**	94.19	0.22	1.34
Average	-	0.14	1.49	62.12	0.20	1.65	126.24	**0.30**	1.71	201.90	0.24	1.52	56.65	0.26	1.70	74.20	**0.30**	**1.75**	**56.58**	0.28	**1.75**

Note: All results of the baselines reported here were obtained by running the official source code at the original experimental settings.

TABLE III
COMPARISON BETWEEN SIX BASELINES AND OUR METHOD IN THE DEFECT IMAGE GENERATION PERFORMANCE IN TERMS OF THE KID$\times 10^3$, LPIPS AND IS VALUES OBTAINED ON THE GDXRAY [53] AND DEEPCRACK [54] DATA SETS.

Data Set	**Crop-Paste** [40]			**DFMGAN** [1]			**AnomalyDiffusion** [21]			**Defect-Gen** [23]			**DefectDiffu** [24]			**SeaS** [25]			**Ours**		
	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑	KID ↓	LPIPS ↑	IS ↑
GDXray [53]	-	0.11	1.35	47.11	0.26	1.97	52.84	0.28	1.96	68.35	0.24	1.85	55.62	0.27	2.92	61.28	0.25	2.88	**43.56**	**0.32**	**2.97**
DeepCrack [54]	-	0.29	1.95	113.97	0.35	2.14	196.60	0.31	1.42	145.23	0.33	1.98	68.45	**0.41**	2.15	97.67	0.32	1.95	**56.50**	**0.41**	**2.24**

TABLE IV
COMPARISON BETWEEN FIVE BASELINES AND OUR METHOD IN TRAINING TIME AND INFERENCE EFFICIENCY.

Method	**Training Time (h)**			**Inference Speed (Pairs/Second)** ↑
	Stage 1	**Stage 2**	**Total**	
DFMGAN	0.87	0.18	1.05	**11.74**
AnomalyDiffusion	1.21	1.39	2.60	0.37
Defect-Gen	0.21	0.14	**0.35**	0.014
DefectDiffu	-	2.15	2.15	0.31
SeaS	16.90	3.46	20.36	0.026
DefectSynth (Ours)	-	1.03	1.03	0.23

generate reasonable road cracks, with the defects produced deviating from the shape characteristics of real road cracks. In addition, DefectDiffu performed relatively well on this data set while SeaS suffered from mask-crack misalignment, where the generated crack regions did not spatially correspond to the indicated mask positions.

3) Computational Efficiency Analysis: To analyze the computational efficiency of the proposed DefectSynth, we assessed the training time and inference speed of five baselines and it on the wood hole category of the MVTec-AD [52] data set. Each model was trained for 50 epochs with a batch size of 1 and an image resolution of 256×256 pixels. During the inference stage, we generated 500 mask-image pairs using the default settings of each model at the resolution of 256×256 pixels. Training time and inference speed are reported in Table IV. It can be seen that our method required a moderate training time and a moderate inference speed. Taking the results reported in Table II, it is demonstrated that our method achieved a reasonable tradeoff between accuracy and efficiency.

C. Ablation Study

To evaluate the effectiveness of each component of our DefectSynth, we conducted an ablation study by removing the component from the method, or replacing it by a different module, in which the same training and testing protocols were used as those described in Section IV-A. For simplicity, only the Wood category in the MVTec-AD [52] data set was used.

1) Effect of the Hybrid Mask Interpolation Module: When the HMI module is included, the model generates intermediate masks through interpolation, enhancing the diversity of reasonable shapes and positions of defects. In this case, a low KID value indicates that the generated image has a high similarity to the authentic defective images. Given that the HMI module was removed from our method, traditional data augmentation operations with random geometric transformations were used to derive more masks. As a result, the highest KID value and the lowest IS value were derived, as shown in Table V, suggesting a low authenticity of the defect shapes. These results show that the HMI module is important for generating diverse and realistic shapes and positions of defects.

To verify the effectiveness of the HMI module, we further compared the performance of four mask generation strategies, including Perlin noise, image-space interpolation only, latent-space interpolation only and the proposed hybrid strategy. In addition to the KID, LPIPS and IS metrics, Mask IoU (higher is better) was utilized. Table VI reports the average

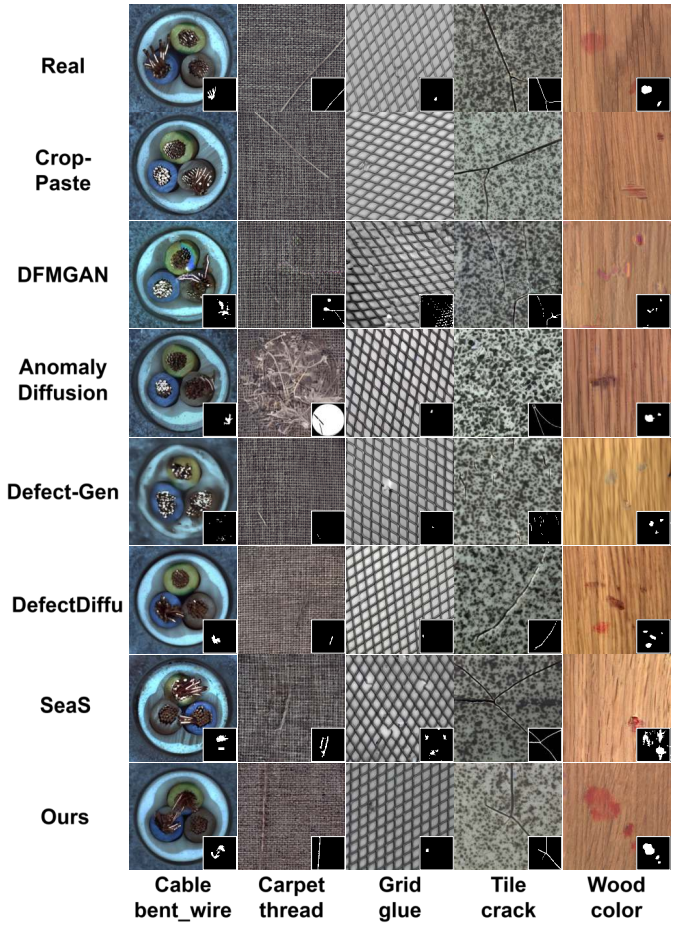

Fig. 3. Comparison of the defective images generated using six baselines and our method on the MVTec-AD [52] data set.

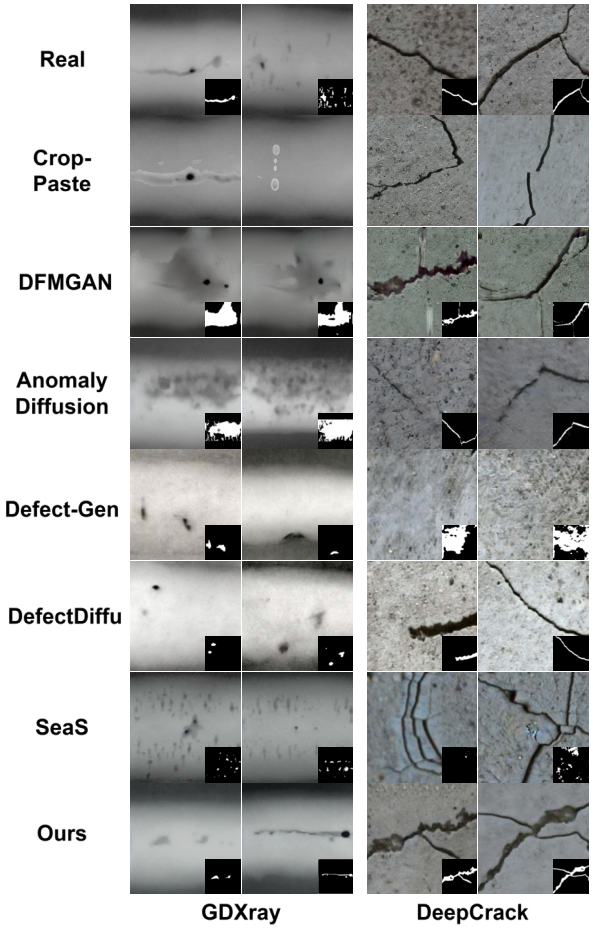

Fig. 4. Comparison of the defective images generated using six baselines and our method on the GDXray [53] and DeepCrack [54] data sets.

TABLE V
THE COMPARISON BETWEEN THE RESULTS OBTAINED USING THREE VARIANTS OF OUR METHOD OBTAINED BY REMOVING A COMPONENT FROM IT AND THOSE DERIVED USING OUR METHOD DIRECTLY.

Method			Metric		
HMI	SFF	SAE	KID ↓	LPIPS ↑	IS ↑
×	✓	✓	127.16	**0.41**	1.82
✓	×	✓	53.61	0.33	1.95
✓	✓	×	61.59	0.35	2.01
✓	✓	✓	**53.40**	0.36	**2.03**

TABLE VI
PERFORMANCE COMPARISON OF FOUR MASK GENERATION STRATEGIES.

Strategy	Mask IoU ↑	KID ($\times 10^3$) ↓	LPIPS ↑	IS ↑
Perlin Noise	0.35	72.63	0.33	1.91
Image-Space-Only	0.47	55.73	0.34	1.98
Latent-Space-Only	0.42	**50.66**	0.31	1.93
Hybrid (Ours)	**0.51**	53.40	**0.36**	**2.03**

performance in terms of the four metrics. It can be seen that our strategy achieved the best result in terms of Mask IoU, LPIPS and IS, and the second-best result with regard to KID. This should be due to the ability of our strategy to adaptively select an interpolation scheme based on defect size, which retains both the fine-detail generation capability of the Image-Space-Only strategy for small defects and the smooth contour generation ability of the Latent-Space-Only strategy for large defects.

2) Effect of the Similarity-Based Feature Fusion Module: The SFF module integrates local features of different images, improving the diversity and visual quality of the defective images generated. Using this module, our method generated more diverse and realistic defective appearances, resulting in higher LPIPS and IS values and a lower KID value. In contrast, the performance dropped without the use of the SFF module, as reported in Table V. This finding highlights the importance of the SFF module in enhancing the diversity of generated images while ensuring their similarity.

3) Effect of the Selective Attention Enhancement Mechanism: The SAE module, which allocates more attention to less prominent defects, aims to improve the structural accuracy of the defects generated. It can be seen from Table V that the model achieved lower KID values together with this module, indicating better similarity to real defects. In the case where the SAE module was removed, the attention of the model was distributed more evenly, resulting in a poorer representation of less prominent defects and higher KID values.

To further assess the SAE module, we measured the inference time when 50 time steps were used. As shown in Table VII, this module used 1.14 more seconds per image,

TABLE VII
COMPARISON OF THE INFERENCE TIME USED BY OUR METHOD WITH AND WITHOUT THE SAE MODULE.

Variant	Time per Image (s)
w/ SAE	3.40
w/o SAE	**2.26**

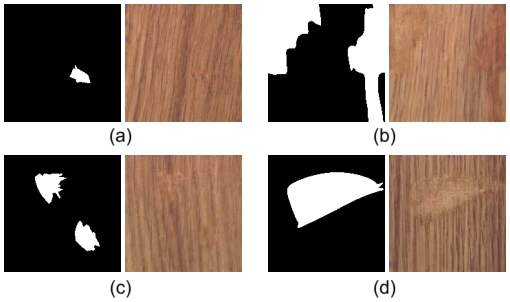

Fig. 5. Three defective images (a-c) generated using three variants of our method obtained by removing the HMI, SFF or SAE module from it, respectively, and a defective image (d) generated by our method directly.

increasing the total generation time by 50%. However, the use of the SAE module yielded a much lower KID value, suggesting a reasonable trade-off between accuracy and efficiency.

4) Summary: As shown in Table V, the proposed DefectSynth, which was built on top of the HMI, SFF and SAE modules, achieved the lowest KID value, the second-highest LPIPS value and the highest IS value, demonstrating the effectiveness of these components. Fig. 5 displays the defective images generated using the three variants of our method obtained by removing the HMI, SFF or SAE module from it, respectively, and that generated by our method directly. As can be observed, the image that our DefectSynth generated shows higher realism and diversity, compared to the three variants. In particular, this image exhibits a more accurate shape, better background integration, and more precise representation of less prominent defects.

V. DOWNSTREAM TASK EXPERIMENTS

To further assess the usefulness of our method, we applied the defective images generated by it to two downstream tasks, including defect classification and localization. Here, the MVTec-AD [52], GDXray [53] and DeepCrack [54] data sets were utilized. The experimental setup and results will be reported as follows.

A. Experimental Setup

1) Evaluation Metrics: For defect classification, we adopted Accuracy (Acc) and F1-Score as evaluation metrics. Accuracy measures the overall classification correctness by calculating the ratio of correctly predicted samples to all test samples. F1-Score provides a balanced assessment between the precision and recall metrics, where precision reflects the ability of a model to avoid false positives while recall indicates the sensitivity of the model to identifying true positives.

Regarding defect localization, we employed three metrics, including Area Under the ROC Curve (AUC), Average Precision (AP) and pixel-level F1-Score. AUC evaluates the discrimination capability of a model between defective and normal pixels across different thresholds. To assess the localization performance at varying recall levels, AP summarizes the precision-recall curve. The pixel-level F1-Score is computed on binarized prediction masks with a fixed threshold (0.5), which directly measures the pixel-wise agreement between the predictions and the ground-truth data.

2) Implementation Details: With regard to the defect classification task on the MVTec-AD [52] data set, we followed the scheme used in [1] by randomly selecting 1/3 of the defective images per category as the base set for training our DefectSynth, which corresponded to 2 to 10 images per category. The remaining 2/3 (6 to 20 images) of the defective images per category were used as the test set for evaluating downstream tasks. Our network and baseline networks were trained using the base set, generating 1,000 images per category. These images were then merged with the base set to form a training set for training a classification model. The ResNet-34 [58] network was used as the classification model, which was trained for 30 epochs with a learning rate of 0.0001. We evaluated the model on the test set without using the augmented data. We repeated the experiment three times with different partitions of the base and test sets and reported the average results. For comparison purposes, we also trained a ResNet-34 network using the base set directly without data augmentation fulfilled by a generation model.

In terms of the defect localization experiment on the MVTec-AD [52], GDXray [53] and DeepCrack [54] data sets, we followed the setup in [21] by selecting the lowest 1/3 of the images by ID from each data set as the base set (i.e., 2 to 10 images for MVTec-AD, 5 images for GDXray and 7 images for DeepCrack). The remaining 2/3 of the images in each data set were used as a test set (i.e., 6 to 20 images for MVTec-AD, 10 images for GDXray and 14 images for DeepCrack). The base set was used to train a defective image generation model. Using a generation model, 500 defective images were generated per data set. To perform the defect localization task on a data set, we trained a U-Net [59] model using the 500 images for 200 epochs with a learning rate of 0.0001, while performing the evaluation on the test set. For comparison purposes, we also trained a U-Net model using the base set directly without data augmentation fulfilled by a generation model.

B. Experimental Results

1) Defect Classification: The results obtained using the ResNet-34 [58] model trained on images generated by the baseline methods and our method are presented in Table VIII. It is evident that classification performance benefits from data augmentation using defective images generated by all methods, confirming the utility of synthetic images for training a more robust ResNet-34 model.

Specifically, DFMGAN [1] achieved good performance on the Hazelnut category but produced low accuracy and F1-Score values on the other categories. AnomalyDiffusion [21]

TABLE VIII
CLASSIFICATION RESULTS OBTAINED USING THE RESNET-34 MODEL, TRAINED ON IMAGES GENERATED BY BASELINE METHODS AND OUR METHOD ON THE MVTEC-AD DATA SET.

Category	**w/o Augmentation** [40]		**DFMGAN** [1]		**AnomalyDiffusion** [21]		**Defect-Gen** [23]		**DefectDiffu** [24]		**SeaS** [25]		**Ours**	
	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑	Acc ↑	F1-Score ↑
Bottle	60.23	53.59	53.46	55.39	64.39	62.59	83.33	82.99	**88.41**	**87.35**	41.27	39.37	88.24	87.32
Cable	36.57	31.12	46.00	39.96	53.46	65.59	**71.43**	**70.05**	57.83	57.83	58.48	51.08	63.37	64.34
Capsule	47.25	41.59	33.50	33.65	**53.73**	**53.69**	36.09	31.16	47.89	50.22	42.94	46.89	53.38	50.30
Carpet	43.62	36.95	50.02	46.53	41.41	34.45	45.56	47.23	62.81	35.84	34.72	37.96	**67.17**	**62.78**
Grid	25.31	19.57	46.10	39.37	50.69	49.55	35.88	38.30	**70.70**	60.64	66.32	**66.86**	65.62	66.79
Hazelnut	51.56	41.67	**83.42**	**79.71**	66.30	61.52	65.22	68.47	78.57	68.46	42.86	37.47	76.54	74.37
Leather	63.53	55.17	46.72	44.03	59.45	54.63	58.22	56.75	64.57	54.47	41.74	39.56	**73.97**	**71.44**
Metal Nut	57.81	53.00	64.77	62.39	74.77	72.67	65.71	68.80	85.70	85.65	55.48	42.98	**91.10**	**91.41**
Pill	38.89	35.23	27.08	27.21	**61.46**	**58.66**	35.82	28.89	50.85	49.66	55.60	56.70	38.65	33.59
Screw	22.12	18.51	45.64	44.00	50.48	55.49	41.27	43.85	**67.50**	**58.37**	46.89	46.23	58.07	55.70
Tile	63.71	55.48	72.79	74.45	84.62	81.52	51.52	35.03	84.05	73.96	38.57	33.71	**88.47**	**90.10**
Toothbrush	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Transistor	23.53	16.68	55.36	**54.75**	47.56	34.96	40.00	37.43	47.28	45.32	32.50	20.74	**57.96**	49.53
Wood	46.06	37.42	50.85	48.44	65.91	62.08	32.00	20.23	68.33	67.98	43.33	31.66	**74.93**	**70.99**
Zipper	36.22	31.35	28.75	27.54	38.05	35.65	25.00	23.71	**52.27**	**48.88**	35.13	33.97	36.28	30.59
Average	44.03	37.67	50.32	48.39	58.02	55.93	49.08	46.63	66.20	60.33	45.42	41.80	**66.70**	**64.23**

TABLE IX
DEFECT LOCALIZATION RESULTS OF THE U-NET MODEL TRAINED ON IMAGES GENERATED BY BASELINE METHODS AND OUR METHOD ON THE MVTEC-AD, GDXRAY AND DEEPCRACK DATA SETS.

Generative Models	MVTec AD [52]						GDXray [53]						DeepCrack [54]					
	Image-level			Pixel-level			Image-level			Pixel-level			Image-level			Pixel-level		
	AUROC ↑	AP ↑	F1 ↑	AUROC ↑	AP ↑	F1 ↑	AUROC ↑	AP ↑	F1 ↑	AUROC ↑	AP ↑	F1 ↑	AUROC ↑	AP ↑	F1 ↑	AUROC ↑	AP ↑	F1 ↑
w/o Augmentation	80.94	70.81	73.30	77.37	58.44	56.06	88.57	86.09	85.74	84.15	71.63	70.48	93.60	91.63	90.21	90.73	70.44	70.08
DFMGAN [1]	91.36	89.54	85.27	90.34	61.14	59.23	98.89	99.09	95.24	94.15	76.42	76.25	**100**	**100**	**100**	98.48	80.80	75.76
AnomalyDiffusion [21]	94.47	91.64	89.21	91.37	63.13	61.95	**100**	**100**	**100**	92.69	73.88	73.20	**100**	**100**	**100**	99.35	78.36	74.71
Defect-Gen [23]	89.47	71.18	70.26	83.46	67.17	66.91	90.53	91.88	89.43	88.20	64.34	62.81	95.69	94.87	94.41	94.34	72.64	71.40
DefectDiffu [24]	95.66	93.38	91.36	**95.42**	75.65	72.10	**100**	**100**	**100**	95.20	82.49	74.73	**100**	**100**	**100**	99.05	83.23	81.36
SeaS [25]	92.37	92.62	90.74	92.97	74.13	70.54	99.39	99.97	97.64	94.20	77.64	72.49	**100**	**100**	**100**	97.38	79.62	77.58
Ours	**96.60**	**93.54**	**92.08**	93.43	**76.05**	**72.32**	**100**	**100**	**100**	**97.46**	**82.56**	**76.64**	**100**	**100**	**100**	**99.76**	**85.48**	**83.27**

performed better than, or comparably to, our method on some categories. For example, it achieved the higher accuracy and F1-Score values than our method on the Capsule, Pill, Screw and Zipper categories. However, its performance was inferior to ours on the other categories, such as Cable, Carpet, Grid and Leather. Defect-Gen [23] and SeaS [25] showed moderate overall performance. Although DefectDiffu [24] demonstrated strong competitiveness in categories such as Bottle, Grid, Screw, and Zipper, its overall performance exhibited notable inconsistency across different categories.

Compared to the baseline methods, our approach achieved superior performance across the majority of the 15 categories. Obvious performance gains were observed, particularly in challenging categories, including Carpet, Leather, Metal Nut, and Wood. These findings demonstrate the consistent effectiveness and superior capability of our method in generating high-quality defective images that robustly enhance classification performance across a diverse range of defect types.

2) Defect Localization: We further evaluated the generated images in the defect localization task using the GDXray [53] and DeepCrack [54] data sets, in addition to MVTec-AD. A U-Net [59] model was trained separately on the images generated by each method for localization. The comprehensive results are summarized in Table IX.

MVTec-AD: For the MVTec-AD data set, our method normally achieved the best performance across the image-level and pixel-level metrics. Specifically, it attained the highest AUROC, AP, and F1-score values at the image level, along with the highest AP and F1-score values at the pixel level.

GDXray and DeepCrack: On both the GDXray and DeepCrack data sets, our method consistently achieved the best performance across all evaluation metrics. The substantial improvement over the baseline method without data augmentation demonstrates the superior capability and strong generalization of our approach for defect localization in diverse inspection scenarios.

Overall Observations: The above results demonstrate the effectiveness of our method in boosting the performance of a defect classification or localization task. Our method was able to generate high-quality, realistic defective images for both defect and texture categories. Even in cases where the defective area was small, it was still capable of capturing the subtle characteristics of defects and presenting them in the images generated.

C. Real-World Industrial Case Study: Tire Defect Inspection

To validate the effectiveness of our method in a real-world industrial inspection scenario, we applied it to tire defect inspection. The model trained using a tire defect data set that

TABLE X
COMPARISON BETWEEN THE RESULTS PRODUCED WITHOUT DATA AUGMENTATION AND AUGMENTED BY OUR DEFECTSYNTH FOR THE DEFECT CLASSIFICATION AND LOCALIZATION TASKS IN A REAL-WORLD TIRE DEFECT INSPECTION SCENARIO.

Task	Metric	w/o Aug. (%)	w/ DefectSynth (%)
Defect Classification	Acc.	78.65	**99.74** (+21.09)
	F1	69.34	**100.00** (+30.66)
Image-Level Localization	AUC	65.79	**94.63** (+28.84)
	AP	78.36	**98.74** (+20.38)
	F1	75.36	**85.21** (+9.85)
Pixel-Level Localization	AUC	68.10	**99.16** (+31.06)
	AP	51.71	**91.37** (+39.66)
	F1	53.62	**88.56** (+34.94)

we derived was used to generate defective tire images. We used these images to train defect classification and localization models. These models were compared with those trained using only real defective images.

1) Tire Defect Data Set: We obtained a real-world tire defect data set from an actual tire production line. The data set contains two common defect categories, i.e., bubbles and foreign objects, which contain 592 and 270 images, respectively. These images were captured using an X-ray device. We annotated each defect identified by inspectors.

2) Experimental Protocol: We followed the same few-shot training and evaluation protocol used in the downstream task (see Section V-A). For each defect category, we used the real training samples to train DefectSynth. The model trained was used to generate synthetic defective images. We then trained a ResNet-34 [58] classifier and a U-Net [59] localizer using these images. The real testing images were used to assess the classifier and localizer.

3) Results: Table X reports the results derived without data augmentation and augmented by our DefectSynth for the defect classification and localization tasks. As can be seen, the data augmentation fulfilled by our method substantially boosted the performance of both tasks regardless of which metric was considered. It is demonstrated that the proposed method is able to overcome the data scarcity issue and improve the accuracy of automated visual inspection approaches in the real-world tire inspection scenario.

VI. CONCLUSION

In this paper, we introduced a novel few-shot defective image generation network, namely, DefectSynth, on top of the diffusion model. This network aims to mitigate the challenge of limited defective samples that industrial inspection normally encounters. DefectSynth explicitly decouples the defect generation process into two stages, including defect mask generation and defective appearance synthesis. We introduced a Hybrid Mask Interpolation (HMI) module in the first stage. This module generates continuous and diverse defect masks using image interpolation or latent interpolation. During the second stage, the pre-trained Stable Diffusion model is used to synthesize realistic and diverse defective images, with the guidance of the fine-tuned ControlNet on top of the mask generated in the first stage and a text prompt. In addition, we adopted a Selective Attention Enhancement (SAE) mechanism and a Similarity-Based Feature Fusion (SFF) module, which serve to enhance the prominence of defects and enrich the textural diversity, respectively. Extensive experiments were conducted on three data sets, including MVTec-AD, GDXray and DeepCrack. The results demonstrated that our approach is able to generate images with high image similarity and diversity, showing a promising advantage. In addition, the performance of defect detection tasks was greatly boosted using the defect data that our method generated. We believe that these promising results should be due to the capability of our DefectSynth to generate diverse and realistic defective images via modeling both the shape and appearance of defects.

REFERENCES

[1] Y. Duan, Y. Hong, L. Niu, and L. Zhang, "Few-shot defect image generation via defect-aware feature manipulation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 1, 2023, pp. 571–578.

[2] W. Cui, K. Song, Y. Zhang, X. Jia, X. Liu, and Y. Yan, "Trdm: A two-stage real-time discrimination method for spiral weld defects under dynamic distorted imaging," *IEEE Transactions on Automation Science and Engineering*, 2025.

[3] Y. Tai, K. Yang, T. Peng, Z. Huang, and Z. Zhang, "Defect image sample generation with diffusion prior for steel surface defect recognition," *IEEE Transactions on Automation Science and Engineering*, 2024.

[4] X. Dong, C. J. Taylor, and T. F. Cootes, "Defect classification and detection using a multitask deep one-class cnn," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1719–1730, 2021.

[5] Y. Du, H. Chen, Y. Fu, J. Zhu, and H. Zeng, "Aff-net: A strip steel surface defect detection network via adaptive focusing features," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[6] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Fully convolutional cross-scale-flows for image-based defect detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 1088–1097.

[7] R. Xu, R. Hao, and B. Huang, "Efficient surface defect detection using self-supervised learning strategy and segmentation network," *Advanced Engineering Informatics*, vol. 52, p. 101566, 2022.

[8] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[9] X. Yan, H. Zhang, X. Xu, X. Hu, and P.-A. Heng, "Learning semantic context from normal samples for unsupervised anomaly detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 4, 2021, pp. 3110–3118.

[10] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep learning for unsupervised anomaly localization in industrial images: A survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–21, 2022.

[11] M. Yang, P. Wu, and H. Feng, "Memseg: A semi-supervised method for image surface defect detection using differences and commonalities," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105835, 2023.

[12] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but differnet: Semi-supervised defect detection with normalizing flows," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1907–1916.

[13] Y. Li, X. Wu, P. Li, and Y. Liu, "Ferrite beads surface defect detection based on spatial attention under weakly supervised learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.

[14] X. Wu, T. Wang, Y. Li, P. Li, and Y. Liu, "A cam-based weakly supervised method for surface defect inspection," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.

[15] V. Zavrtanik, M. Kristan, and D. Skočaj, "Draem-a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 8330–8339.

[16] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9664–9674.

[17] G. Zhang, K. Cui, T.-Y. Hung, and S. Lu, "Defect-gan: High-fidelity defect synthesis for automated defect inspection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2524–2534.

[18] Z. Du, L. Gao, and X. Li, "A new contrastive gan with data augmentation for surface defect recognition under limited data," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2022.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[21] T. Hu, J. Zhang, R. Yi, Y. Du, X. Chen, L. Liu, Y. Wang, and C. Wang, "Anomalydiffusion: Few-shot anomaly image generation with diffusion model," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 8, 2024, pp. 8526–8534.

[22] Q. Yu, K. Zhu, Y. Cao, F. Xia, and Y. Kang, "Tf 2: Few-shot text-free training-free defect image generation for industrial anomaly inspection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[23] S. Yang, Z. Chen, P. Chen, X. Fang, Y. Liang, S. Liu, and Y. Chen, "Defect spectrum: A granular look of large-scale defect datasets with rich semantics," in *European Conference on Computer Vision*. Springer, 2024, pp. 187–203.

[24] Q. Shi, J. Wei, F. Shen, and Z. Zhang, "Few-shot defect image generation based on consistency modeling," in *European Conference on Computer Vision*. Springer, 2024, pp. 360–376.

[25] Z. Dai, S. Zeng, H. Liu, X. Li, F. Xue, and Y. Zhou, "Seas: Few-shot industrial anomaly image generation with separation and sharing fine-tuning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 23 135–23 144.

[26] D. P. Kingma, M. Welling *et al.*, "Auto-encoding variational bayes," 2013.

[27] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[28] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal difference variational auto-encoder," *arXiv preprint arXiv:1806.03107*, 2018.

[29] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[30] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[31] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[32] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[33] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[34] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[35] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3836–3847.

[38] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2426–2435.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[40] D. Lin, Y. Cao, W. Zhu, and Y. Li, "Few-shot defect segmentation leveraging abundant defect-free training samples through normal background regularization and crop-and-paste operation," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[41] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated defect inspection system for metal surfaces based on deep learning and data augmentation," *Journal of Manufacturing Systems*, vol. 55, pp. 317–324, 2020.

[42] S. Wang, Z. Zhong, Y. Zhao, and L. Zuo, "A variational autoencoder enhanced deep learning model for wafer defect imbalanced classification," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 12, pp. 2055–2060, 2021.

[43] H. Fu and F. Kang, "Self-attention adversarial variational autoencoders networks for rail surface defect data expansion," *Engineering Research Express*, vol. 7, no. 4, p. 0452f3, 2025.

[44] Y. Wang, W. Hu, L. Wen, and L. Gao, "A new foreground-perception cycle-consistent adversarial network for surface defect detection with limited high-noise samples," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11 742–11 751, 2023.

[45] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with gan for improving defect recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1611–1622, 2020.

[46] H. Ahmad, A. Banjar, A. O. Alzahrani, I. Ahmad, M. S. Naeem *et al.*, "Image synthesis of apparel stitching defects using deep convolutional generative adversarial networks," *Heliyon*, vol. 10, no. 4, 2024.

[47] X. Jiang, Y. Li, F. Yan, Y. Lu, C. Xu, and M. Xu, "Mgdefect: A mask-guided high-quality defect image generation method for improving defect inspection," *IEEE Transactions on Multimedia*, 2025.

[48] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[49] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 245–254.

[50] K. Zhang, Y. Zhou, X. Xu, B. Dai, and X. Pan, "Diffmorpher: Unleashing the capability of diffusion models for image morphing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7912–7921.

[51] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[52] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.

[53] D. Mery, V. Riffo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, H. Lobel, and M. Carrasco, "Gdxray: The database of x-ray images for nondestructive testing," *Journal of Nondestructive Evaluation*, vol. 34, no. 4, p. 42, 2015.

[54] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.

[55] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401*, 2018.

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[57] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.