FabGPT: An Efficient Large Multimodal Model for Complex Wafer Defect Knowledge Queries

Yuqi Jiang,   Xudong Lu,   Qian Jin,   Qi Sun[#],   Hanming Wu,   Cheng Zhuo[#]

Zhejiang University

ABSTRACT

Intelligence is key to advancing integrated circuit (IC) fabrication. Recent breakthroughs in Large Multimodal Models (LMMs) have unlocked unparalleled abilities in understanding images and text, fostering intelligent fabrication. Leveraging the power of LMMs, we introduce FabGPT, a customized IC fabrication large multimodal model for wafer defect knowledge query. FabGPT manifests expertise in conducting defect detection in Scanning Electron Microscope (SEM) images, performing root cause analysis, and providing expert question-answering (Q&A) on fabrication processes. FabGPT matches enhanced multimodal features to automatically detect minute defects under complex wafer backgrounds and reduce the subjectivity of manual threshold settings. Besides, the proposed modulation module and interactive corpus training strategy embed wafer defect knowledge into the pre-trained model, effectively balancing Q&A queries related to defect knowledge and original knowledge and mitigating the modality bias issues. Experiments on in-house fab data (SEM-WaD) show that our FabGPT achieves significant performance improvement in wafer defect detection and knowledge querying.

1 INTRODUCTION

The intersection of visual and language models [1–3] has significantly propelled the revolutionary advancement of artificial intelligence (AI), which makes models understand and interpret the world similarly to humans. Since Large Multimodal Models (LMMs) [4–6] possess the capability to reason about visual images, they have attracted considerable attention in defect detection tasks. However, current LMMs are primarily applied to visual tasks [7–9] in basic scenarios and lack sensitivity to the knowledge of specialized domains. This limits their efficiency in wafer defect knowledge query in the field of integrated circuits (IC) fabrication.

In the semiconductor industry, the manufacturing process is intricate, with each step potentially introducing random defects. These defects impact the reliability of electronic devices [10–13], making it essential to detect defects on the wafer surface and perform thorough question-and-answer (Q&A) analysis to deepen engineers' understanding of these defects and IC questions.

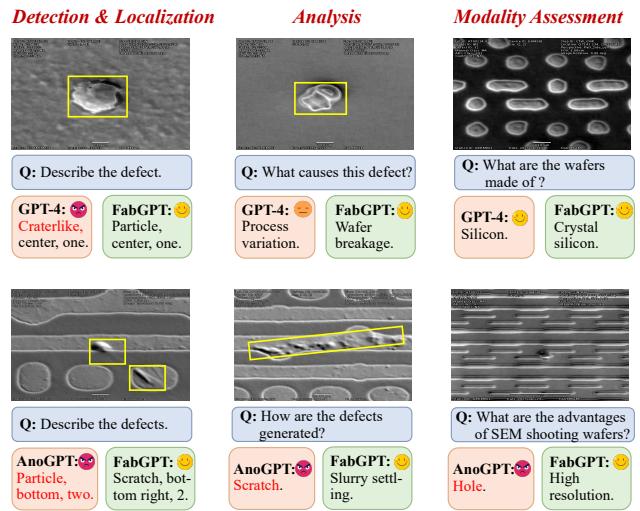[#] Corresponding authors: {qisunchn, czhuo}@zju.edu.cn.

Figure 1: Comparisons between our FabGPT and GPT-4 [1], AnomalyGPT [19], which are fine-tuned on our dataset, for detecting, locating, and analyzing microscopic defects in complex backgrounds and addressing modality bias issues. Previous arts perform badly while encountering "detection", "analysis", and "modality bias".

Recent years have witnessed advancements in methods for querying defect knowledge, encompassing both detection and Q&A analysis. Convolutional Neural Networks (CNNs)-based approaches [14–18] leverage extensive training data to recognize patterns and features in images, thereby enhancing the accuracy and efficiency of defect detection. However, these methods are heavily dependent on large-scale annotated data and fall short of conducting in-depth Q&A analysis, which limits their ability to understand complex image content. Moreover, some methods [19, 20] employ large models, fine-tuning pre-trained models with specific data sets to achieve superior detection performance, even with limited data availability. These approaches also demonstrate strong visual understanding capabilities, enabling them to deduce relevant visual knowledge. Consequently, integrating wafer defect knowledge from the IC domain into large models is promising to support comprehensive defect knowledge queries.

However, existing methods based on fine-tuning large models often face two issues: 1) difficulty in precisely detecting defects within complex backgrounds; and 2) a loss of the ability to perform comprehensive Q&A tasks. Despite the vast amount of knowledge stored in large models, the complex and microscopic structure of wafer surfaces still prevents them from accurately capturing information such as the number and location of minute defects. As shown in Figure 1, GPT-4 [1] struggles with queries specific to

wafer defects, and even the fine-tuned AnomalyGPT [19] inaccurately identifies the locations of minute defects. Additionally, as demonstrated by the dialogue in Figure 1 with AnomalyGPT [19], the fine-tuned model tends to produce text outputs that are still biased towards visual content when user queries are not closely related to the visual input. This is termed as "modality bias", indicating that the model loses its ability to judge and understand the textual content of the questions.

To address the issues mentioned above, we propose an efficient LMM, FabGPT, that employs a three-stage strategy: modal enhancement, detection, and Q&A stage. This strategy allows us to build on the inherent capabilities of pre-trained models by embedding high-quality prompt instructions, enabling it to detect wafer defects in the IC domain and query the related knowledge. Additionally, to further alleviate "modality bias", we introduce an interactive corpus training strategy.

Our contributions are summarized as follows:

- We propose a knowledge query LMM, FabGPT, based on prompt learning, which effectively detects minute defects in complex wafer backgrounds and conducts Q&A analysis on relevant defect knowledge.
- We design a modal enhancement stage that constrains and supplements the semantic information in multimodal features, significantly optimizing the quality of the prompt features.
- A detection head is developed, matching multimodal features to automatically detect pixel-level defects. This device eliminates the drawbacks of subjective threshold selection while enhancing defect detection capabilities.
- We propose a Q&A stage and a corpus training strategy that supervises real-time updating of instruction coefficients and the interaction of new and old knowledge, addressing the modality bias in dialogues.
- We conduct comprehensive experiments on our SEM-WaD dataset, our FabGPT achieves the supervised detection accuracy of 91.81% image-level, 95.61% pixel-level, 88.17% PRO, and 85.80% AP. For Q&A dialogue, it achieves 96.86% accuracy, outperforming the baselines significantly.

2 PRELIMINARIES

2.1 IC Wafer Defect Analysis

Wafers are the fundamental material for manufacturing IC, and the quality of their surface directly impacts the reliability of the chips. By identifying and understanding the various defects on the wafer surfaces detected by Scanning Electron Microscopes (SEM) [10, 11, 13], engineers can learn about the attributes such as type, location, and cause of the defects, which is crucial for optimizing processes and quality control. Traditional and CNN-based analysis methods utilize large datasets for feature learning to detect defect regions. For example, Zontak *et al.* [21] utilized the periodicity of wafer patterns to manually construct defect features, thus detecting defects. Gomez *et al.* [22] proposed a detection method based on Support Vector Machines (SVM), which separates data points of different classes in high-dimensional space using defined hyperplanes. Cheon *et al.* [15] proposed a CNN model that can extract effective features for defect classification. These models achieve significant progress in classifying and segmenting wafer defects. However, they perform
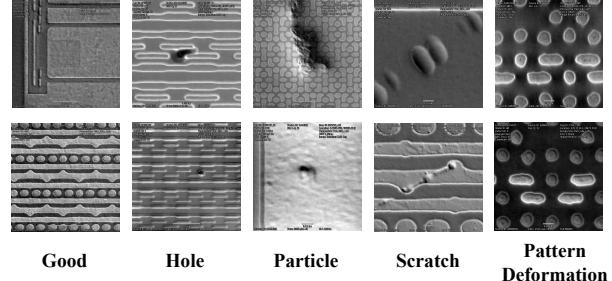

Figure 2: The four types of defects and defect-free (good) images in the SEM-WaD dataset.

poorly when faced with scarce data and have not yet developed a model that integrates classification, segmentation, and analysis. There is an urgent need for an automated tool capable of detecting and inferring wafer defects with minimal data.

2.2 Collection and Processing of Dataset

Due to the absence of publicly available datasets for wafer surface defects, we collect an in-house dataset (SEM-WaD) using SEM techniques from various IC manufacturing steps and products. It comprises 1,226 defect-free images and 1,182 images with four common types of defects (holes: 250, particles: 500, pattern deformities: 250, scratches: 182) from our fab partners. The wafer images in SEM-WaD contain diverse and complex backgrounds, with each defect type exhibiting unique morphologies and features. Consequently, we meticulously annotate each image in the dataset with details such as image IDs and production steps. Some examples are shown in Figure 2.

2.3 Large Multimodal Models

LMMs [4–6] are the large and complex models capable of processing various types of data (images, spectra, sound, *etc.*) and utilize large-scale datasets during training to understand and generate descriptions of visual content. Their powerful comprehension and transfer abilities make them excel in various tasks such as image description generation and visual question answering. For example, [4] utilized a linear layer for aligning the frozen video encoder of the BLIP-2 [23] and the LLM Vicuna [24] to enable image-text Q&A. [5] adopted a contrastive learning approach to embed the semantic information of images and text into the same space to achieve zero-shot transfer. DALL-E [6] generated images related to text descriptions by encoding text with Transformer and using a Generative Adversarial Network (GAN). These models have a strong ability to understand complex image-text data, however, they face challenges in robustness when adapting to new domains.

2.4 Fine-Tuning Methods

Fine-tuning methods primarily include full [25, 26] and partial fine-tuning [19, 20, 27], which involves retraining parts of a model pre-trained on large datasets to adapt to specific tasks. In full fine-tuning, all pre-trained model layer weights are trainable, enabling flexible adaptation to new data features by adjusting all parameters. For example, BERT [25] is trained on large corpora adjusting all parameters in its bidirectional Transformer structure to optimize tasks such as sentiment analysis question answering and text
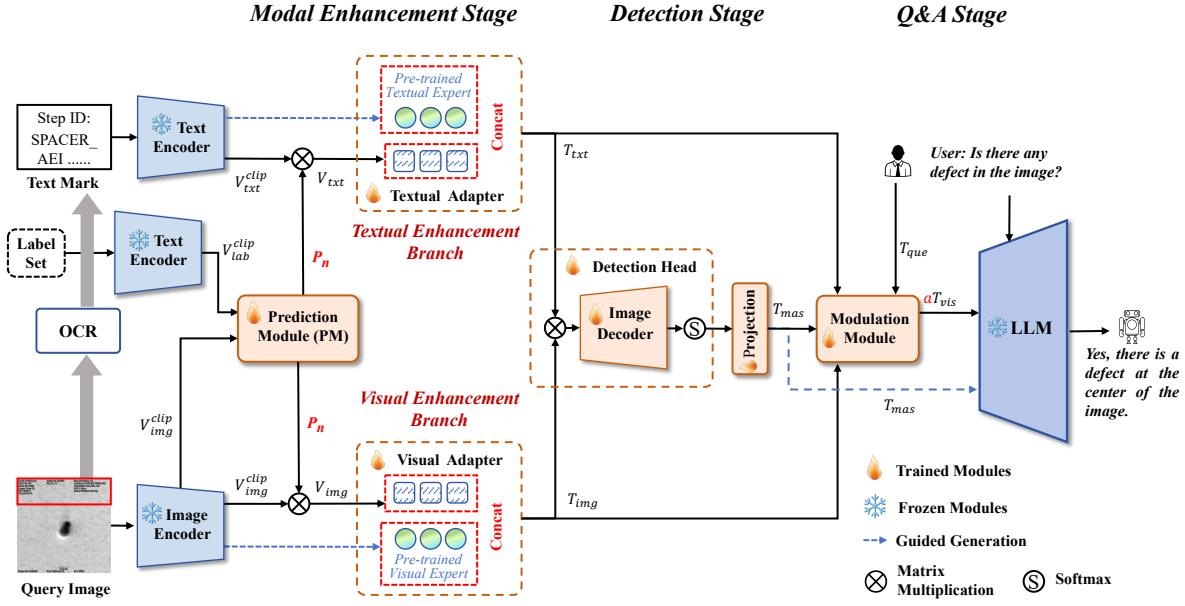
Figure 3: The architecture of FabGPT. The images and the characters extracted from them serve as the primary multimodal input into a three-stage model, with the label set entering as auxiliary textual input. The first stage enhances semantic information of multimodal features. Based on the first stage, the detection stage performs pixel-level automated detection, and the Q&A stage achieves complete Q&A integrating both old and new knowledge.

summarization. ViT [26] adjusts its Transformer structure by specialized visual datasets, efficiently adapting to tasks such as image classification and object detection. Full fine-tuning enhances model adaptation to specific downstream tasks but also poses a higher risk of overfitting and increasing computational costs, limiting its wide applications.

In partial fine-tuning [19, 20, 27], it adjusts only partial parameters in pre-trained models and keeps the rest fixed. Recent popular partial fine-tuning strategies include Adapters, Prompt Learning, and LoRA. For example, Rebuffi *et al.* [27] introduced residual adapters to neural networks, optimizing only these modules to reduce fine-tuning costs due to their fewer parameters. Lai *et al.* [20] applied the LoRA method to consistently update the bottom embeddings and top linear head of the pre-trained model, while randomly updating a few intermediate self-attention layers to understand input text for precise segmentation. Gu *et al.* proposed AnomalyGPT [19], which fine-tunes an LLM using embedded prompt instructions to identify types and locations of defects.

Despite their advancements, new models often lose their normal Q&A capabilities after fine-tuning. This occurs because the model becomes excessively focused on image inputs while neglecting user queries, whether or not these queries are related to the images. This phenomenon is termed "***modality bias***". Our model effectively integrates complex wafer defect knowledge and is designed to alleviate the modality bias.

3 PROPOSED METHOD

3.1 Network Architecture

As shown in Figure 3, our FabGPT is a conversational LMM designed for querying wafer defect knowledge, and it consists of a foundational stage for modal enhancement and two functional stages for detection and Q&A.

Given a query image $x \in \mathbb{R}^{H \times W \times 3}$, text marks are extracted from x using Optical Character Recognition (OCR) technology [28]. The image x, its text marks, and the label set are encoded into initial vectors V_{img}^{clip}, V_{txt}^{clip}, and V_{lab}^{clip} through pre-trained image and text encoders [29]. V_{img}^{clip} and V_{lab}^{clip} are fed into the Prediction Module (PM) to predict the defect category P_n. Then, P_n is used to multiply with V_{img}^{clip} and V_{txt}^{clip}, generating the vectors V_{img} and V_{txt}. Visual and textual adapters further process these vectors into the information-rich image token T_{img} and text token T_{txt}. In the detection stage, T_{img} and T_{txt} are fed into the detection head to obtain supervised detection masks. In the Q&A stage, the Modulation Module aligns T_{img}, T_{txt}, mask-projected token T_{mas}, and the user's question token T_{que} into a unified visual token aT_{vis}. Finally, it is concatenated with T_{mas} and T_{que} to serve as prompt instructions for fine-tuning PandaGPT [2].

3.2 Modal Enhancement Stage

When analyzing the minor defects on the wafer surface, distinguishing between complex background and foreground features is challenging, which hinders queries on defect-related knowledge. To address this, we develop the modal enhancement stage, consisting of the PM and two enhancement branches, designed to highlight relevant defect features and minimize the impact of irrelevant features.

Prediction Module (PM):
The pre-trained encoder captures pixel-level detail features in the latent space, but its repeated down-sampling operations result
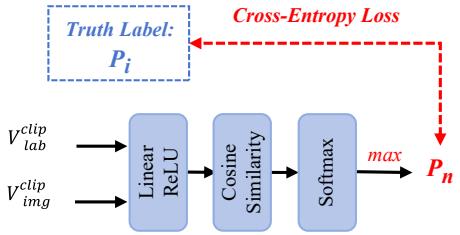
Figure 4: The Prediction Module (PM).

Figure 5: The Modulation Module.

in the loss of semantic features. As shown in Figure 4, we design the PM to predict defect categories in the image and enhance semantic features by embedding the expected result p_n into the initial vectors.

Specifically, we count a label set containing all defect categories found in wafer images to support automated classification tasks. First, the linear layer and the activation function are applied to reshape the dimensions of V_{img}^{clip} and V_{lab}^{clip}, formulated as:

$$f_{img} = \sigma(W_i^T V_{img}^{clip} + b_i),$$
$$f_{lab} = \sigma(W_i^T V_{lab}^{clip} + b_i), \tag{1}$$

where W_i represents the weight matrix, b_i represents the bias vector, and σ represents the ReLU function. Then, the cosine of the angle between each category f_{lab}^i and f_{img} is computed to assess their similarity, and the cross-entropy loss function is used to constrain the selection of the corresponding category p_n. This process is formulated as:

$$p_i = \text{Cosine}(f_{img}, f_{lab}),$$
$$P_n = max(Softmax(p_i)),$$
$$= max(\frac{exp(p_i)}{\sum_{j=1}^{N} exp(p_j)}), \tag{2}$$

where Cosine represents the cosine similarity calculation. p_n is matrix-multiplied with V_{img}^{clip} and V_{lab}^{clip}, resulting in vectors V_{img} and V_{txt}, which are enriched with semantic features.

Two Enhancement Branches:

Although semantic features related to defect attributes are enhanced, the detailed representation of defect features remains essential. In the visual and textual enhancement branches, adapters based on prompt learning are deployed, utilizing the extra prompts of pre-trained experts for adaptive feature optimization. The experts are initialized under the guidance of V_{img}^{clip} and V_{txt}^{clip}, which enable them to acquire knowledge from image and text modals. During training, they adaptively update parameters and interact with V_{img} and V_{txt} after each update, effectively controlling the direction and quality of the feature flow. The final outputs T_{img} and T_{txt} are generated, with T_{img} being represented by the following formula (the same applies to T_{txt}):

$$\nabla f_e = (V_{img}^{clip})^i \rightarrow z^i,$$
$$T_{img} = Concat(\nabla f_e, V_{img}), \tag{3}$$

where $\rightarrow$ represents the feature-guiding operation, z^i and $(V_{img}^{clip})^i$ are the i-th elements of the random vector z and V_{img}^{clip} respectively, and ∇f_e is the prompt feature of the pre-trained expert.
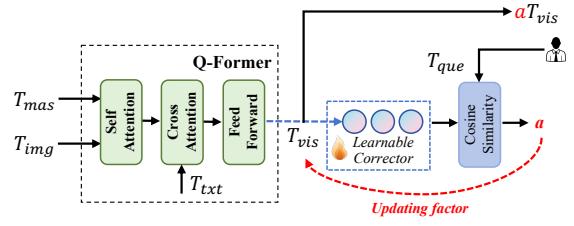
3.3 Detection Stage

Manually setting segmentation thresholds is not necessarily optimal while continuous adjustments should be made for specific tasks. We design the detection head that autonomously learns the specific thresholds for each pixel at feature positions to generate pixel-level masks Mask $\in \mathbb{R}^{H \times W}$. It first fuses complementary information from T_{img} and T_{txt} through matrix multiplication, then maps them back to high-dimensional features through four up-sampling operations of the trainable decoder, and normalizes the output into the mask image through the softmax function. The detection head matches multimodal information and supervises detail features, enabling precise defect detection in complex wafer backgrounds. This process is formulated as:

$$\text{Mask} = Softmax((T_{img} \otimes T_{txt}) \uparrow_s^4), \tag{4}$$

where $(\cdot) \uparrow_s^4$ represents performing four times up-sampling.

3.4 Q&A Stage

In fine-tuning the large models based on prompt instructions, the commonly used embedded instruction format is:

$$\text{INS} = Concat(x, T_{img}, T_{que}). \tag{5}$$

However, this embedding format may lead to modality bias, where the model is dominated by image inputs and fails to respond to questions appropriately. For example, the latest defect Q&A model AnomalyGPT [19] embeds industrial defect knowledge into its pre-trained model based on Equation (5), it can only answer questions related to defects, such as:

- "What's this in the image?"
- "Is there a defect in the image?"
- "Where is the defect located in the image?"

However, it fails to answer general questions not closely related to the images, such as:

- "What are the types of industrial products?"
- "What impact does this defect have on the production line?"
- "What are the core process steps in IC manufacturing?"

This phenomenon indicates that while the model gains new information, it loses the understanding of its original knowledge, thereby diminishing its ability to analyze general knowledge effectively.

We suggest that this phenomenon occurs because the model fails to judge the correlation between the query image and the user's question adequately. The keys to resolving this issue and ensuring accurate model responses are: 1) Improving the quality of visual instructions; 2) Enhancing the ability to assess the relevance between visual prompt instructions and user query ones; 3) Optimizing the training strategies of the corpus.

Modulation Module:

The quality of prompt instructions significantly affects the model's ability to understand knowledge and respond to queries. Thus, it is necessary to align enhanced multimodal features to capture visual information and alleviate the model's fine-tuning burden. Inspired by Q-Former [23], a bidirectional self-attention in Figure 5 allows T_{img} to absorb semantic and detailed information from T_{mas} to obtain $f_{i \sim m}$, facilitating interaction within the same modality, formulated as:

$$M_{img} = Softmax(((T_{img} * k_1)(T_{img} * k_2)^T)/\sqrt{d_k}),$$
$$M_{mak} = Softmax(((T_{mas} * k_1)(T_{mas} * k_2)^T)/\sqrt{d_k}), \quad (6)$$
$$f_{i \sim m} = \frac{(S_i + S_m)}{2} T_{img},$$

where $*$ represents convolution operations, k_i represents different kernels, and d_k represents the dimension of the feature vector. Next, aligning the fine-grained information between visual features $f_{i \sim m}$ and textual tokens T_{txt} through cross-attention [30] allows for the sharing of complementary knowledge across multimodalities, the result $f_{i \sim m \sim t}$ can be formulated as:

$$M_{i \sim m \sim t} = Softmax(((f_{i \sim m} * k_1)(T_{txt} * k_2)^T)/\sqrt{d_k}),$$
$$f_{i \sim m \sim t} = M_{i \sim m \sim t} T_{txt}. \quad (7)$$

Finally, to maintain semantic consistency between the LLM and outputs of the modal enhancement stage, the feed-forward network maps the unified features $f_{i \sim m \sim t}$ to a high-dimensional space and outputs high-quality prompt instructions T_{vis} through the activation of nonlinear layers.

Since the content of user queries involves knowledge of different tasks, we must assess the relationship between query and visual instructions before fine-tuning the LLM. We set a scaling factor a, dynamically adjusting its value through learning the association between instructions (the higher the value, the stronger the association). A learnable corrector that is generated under the guidance of T_{vis} is introduced, and its similarity score with the query is calculated to simulate the value of a. It can be formulated as:

$$a = \frac{\nabla f_c \cdot T_{que}}{||\nabla f_c|| \cdot ||T_{que}||}, \quad (8)$$

where $\cdot$ represents the dot product and ∇f_c represents the prompt features of the learnable corrector. We assign a as the coefficient to T_{vis}. The updated aT_{vis} along with T_{mas} and T_{que} serve as our input instructions, formatted as follows:

$$\widetilde{INS} = Concat(aT_{vis}, T_{mas}, T_{que}). \quad (9)$$

Corpus Training Strategy:

During the corpus training process, the alternating training strategy balances learning new and old knowledge. We establish two corpora: Corpus-A, which includes 15 Q&A pairs for each category related to defect type, quantity, location, description, and analysis (e.g., Q: What type of defect is in the image? A: The defect in the image is object.), and Corpus-B, which contains 100 Q&A pairs unrelated to defect knowledge (e.g., Q: What is the capital of China? A: The capital of China is Beijing.). Our model trains alternately on these corpora at a 2:1 ratio to prevent it from favoring the retrieval of new knowledge when understanding questions.

3.5 Loss Functions

We employ three loss functions to constrain the detection and dialogue processes. Focal Loss [31] and Dice Loss [32] are used to improve the model's segmentation and localization abilities, and Cross-Entropy Loss is used to improve the model's classification and Q&A ones.

Focal Loss:

Focal Loss [31] aims to address the issue of class imbalance. It introduces a modulation factor γ to reduce the relative loss of correctly classified pixels and focus on hard-to-classify and misclassified pixels. It is realized as Equation (10):

$$L_{focal} = -\frac{1}{H \times W} \sum_{i=1}^{H \times W} (1 - p_i)^\gamma \log(p_i), \quad (10)$$

where p_i represents the probability of the pixel belonging to the correct category, and based on [19], we set the γ to 2.

Dice Loss:

Dice Loss [32] aims to maximize the overlap between outputs and actual labels, encouraging the model to learn to produce results closer to the ideal segmentation. It is realized as Equation (11):

$$L_{dice} = -\frac{\sum_{i=1}^{H \times W} y_i \hat{y}_i}{\sum_{i=1}^{H \times W} y_i^2 + \sum_{i=1}^{H \times W} \hat{y}_i^2}, \quad (11)$$

where y_i is the output of the decoder, and $\hat{y}_i$ is the truth labels.

Cross-Entropy Loss:

Cross-entropy loss measures the difference between predicted and actual categories in the PM and between output texts of the language model and target texts in the Q&A task. It is realized as Equation (12):

$$L_{ce} = -\sum_{i=1}^{c} y_i \log(p_i), \quad (12)$$

where c represents the number of categories and tokens in classification and Q&A tasks, y_i represents the truth label and p_i represents the predicted label.

The overall loss function is:

$$L = \alpha L_{focal} + \beta L_{dice} + \delta L_{ce}^1 + \epsilon L_{ce}^2. \quad (13)$$

We set the coefficients α, β, δ and ϵ to 1, by default.

4 EXPERIMENTS

4.1 Experimental Setups

Datasets:

Our experiment is conducted on the in-house SEM-WaD dataset. The dataset comprises images with a resolution of 480×480, each accompanied by a corresponding mask image and related textual descriptions and analyses. We divide the data into training and test sets in a 7:3 ratio, where both sets consist of good and defective images.

Implementation Details:

Our model uses PandaGPT [2] as the foundational LMM, composed of the Vicuna-7B [24] as the language model and ImageBind Huge [29] as the frozen encoder. During training, we employ the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [33] with an initial learning rate of $1e^{-4}$, gradually reducing to $1e^{-6}$, using the cosine annealing

Table 1: The results of the Image-AUC and Pixel-AUC metrics from fully supervised experiments on the SEM-WaD dataset. The best and second-best methods are highlighted in red and blue fonts, respectively.

Category	DevNet [14]	DRA [16]	BGAD [18]	PRN [17]	Lisa [20]	AnomalyGPT [19]	FabGPT (ours)
Hole	63.92 / 80.10	81.44 / 85.75	35.78 / 87.10	79.26 / 84.30	87.37 / 84.31	91.37 / 93.68	94.28 / 97.03
Particle	62.20 / 86.30	96.71 / 96.86	81.87 / 92.75	80.41 / 87.60	93.34 / 91.73	92.64 / 94.39	94.43 / 97.30
Scratch	19.84 / 76.74	89.94 / 75.28	65.27 / 88.98	76.93 / 77.02	84.45 / 85.28	89.58 / 92.58	90.32 / 95.70
Pattern Deformation	65.02 / 48.78	85.52 / 90.71	72.37 / 81.90	75.87 / 76.53	85.51 / 87.72	86.23 / 89.66	88.19 / 92.40
Average	52.74 / 72.98	88.14 / 87.15	63.82 / 87.68	78.12 / 81.36	87.67 / 87.26	89.96 / 92.58	91.81 / 95.61

Table 2: The results of the PRO and AP metrics from fully supervised experiments on the SEM-WaD dataset. The best and second-best methods are highlighted in red and blue fonts, respectively.

Category	DevNet [14]	DRA [16]	BGAD [18]	PRN [17]	Lisa [20]	AnomalyGPT [19]	FabGPT (ours)
Hole	- / 86.98	57.81 / -	80.81 / 51.33	80.90 / 76.48	88.66 / 79.18	86.39 / 83.47	90.01 / 85.69
Particle	- / 92.38	66.67 / -	89.58 / 51.33	88.84 / 84.79	83.57 / 72.87	89.74 / 86.02	92.28 / 91.58
Scratch	- / 50.55	25.83 / -	75.33 / 52.28	67.82 / 51.70	57.77 / 35.73	80.32 / 60.32	83.00 / 79.19
Pattern Deformation	- / 82.07	60.71 / -	81.08 / 54.75	74.63 / 70.57	77.75/ 69.01	84.35 / 80.73	87.40 / 86.72
Average	- / 77.80	52.76 / -	81.70 / 52.66	78.05 / 70.89	76.94/ 64.20	85.20 / 77.64	88.17 / 85.80

[34] strategy. The model is trained on three 4090Ti GPUs, with a batch size of 24 and an epoch of 50.

Evaluations:

In the detection task, we evaluate model performance using four metrics: Image-AUC (the Area Under the Receiver Operating Characteristic Curve), Pixel-AUC, Per-Region Overlap (PRO), and Average Precision (AP). Image-AUC and Pixel-AUC assess the model's ability to judge the presence of defects in images, while PRO and AP measure the precision of the model in identifying and locating defects. In the Q&A task, we conduct 15 questions for each of the four defect types, including inquiries about the presence, category, location, quantity, appearance description, and root cause analysis of defects. Additionally, we pose 50 questions that are unrelated to defects and not included in the corpus (IC-related or IC-unrelated general questions) to validate the modality bias issue. We use the percentage of correct answers as a relevant metric to evaluate the Q&A capability of the model. To demonstrate the outstanding performance of our FabGPT, we compare it with many representative previous arts:

- DevNet [14]: Learns anomaly representations by labeled anomalies and prior probabilities.
- DRA [16]: A CNN-based learning model for detecting anomalies in a composite feature space.
- BGAD [18]: An anomaly scoring model that utilizes explicit boundary generation and boundary-guided optimization.
- PRN [17]: A residual detection model that outputs anomalies by learning different block features.
- Lisa [20]: An anomaly segmentation LMM utilizing LoRA fine-tuning method.
- AnomalyGPT [19]: An anomaly detection LMM utilizing prompting learning fine-tuning method.

These baselines are also tuned on our SEM-WaD dataset, following their publicly available implementations and models. Among these baselines, Lisa and AnomalyGPT are LMMs that support Q&A while the others do not.

4.2 Quantitative Results

The quantitative results report the accuracy of the supervised defect detection and the correctness of the knowledge Q&A responses in detail.

Defect Detection Task:

Table 1 and Table 2 report the Image-AUC, Pixel-AUC, PRO, and AP values for different methods across 4 categories within the SEM-WaD dataset. It can be observed that our model outperforms all other methods in four evaluation metrics for most defect categories. For example, compared to AnomalyGPT [19], which also employs prompt learning for fine-tuning, our model achieves a higher Image-AUC by 1.85% and a higher Pixel-AUC by 3.03%. Compared to the traditional detection model PRN [17], our model surpasses it by 10.12% in PRO and 14.91% in AP.

Q&A Task:

Table 3 reports the accuracy of different language models in answering defect-related and -unrelated questions where our model achieves state-of-the-art results. For example, compared to the powerful capabilities of GPT-4, our model achieves a comparable performance in the general questions and a 7.50% higher accuracy in defect analysis answers. Compared to Lisa [20], which embeds new tasks to pre-trained models, our model achieves a 78.00% higher accuracy in answering questions unrelated to the new task.

4.3 Qualitative Results

We produce comparison figures for detection results and dialogue diagrams for Q&A results on the SEM-WaD dataset. Intuitive detection results are validated using heatmaps and mask images to demonstrate the effectiveness of our model in the supervised detection task.

Defect Detection Task:

Table 3: The accuracy (%) in the models' responses to different questions. The best and second-best methods are highlighted in red and blue fonts, respectively. "-" denotes the question is unsupported.

Methods	Defect-Related						Unrelated	All
	Presence	Category	Location	Quantity	Description	Analysis		
Lisa [20]	95.00	92.50	-	-	72.50	-	20.00	70.00
AnomalyGPT [19]	95.00	75.00	72.50	77.50	82.50	-	12.00	69.08
GPT-4 [1]	100.00	37.50	87.50	80.00	85.00	90.00	98.00	82.57
FabGPT (ours)	100.00	97.50	95.00	95.00	95.00	97.50	98.00	96.86

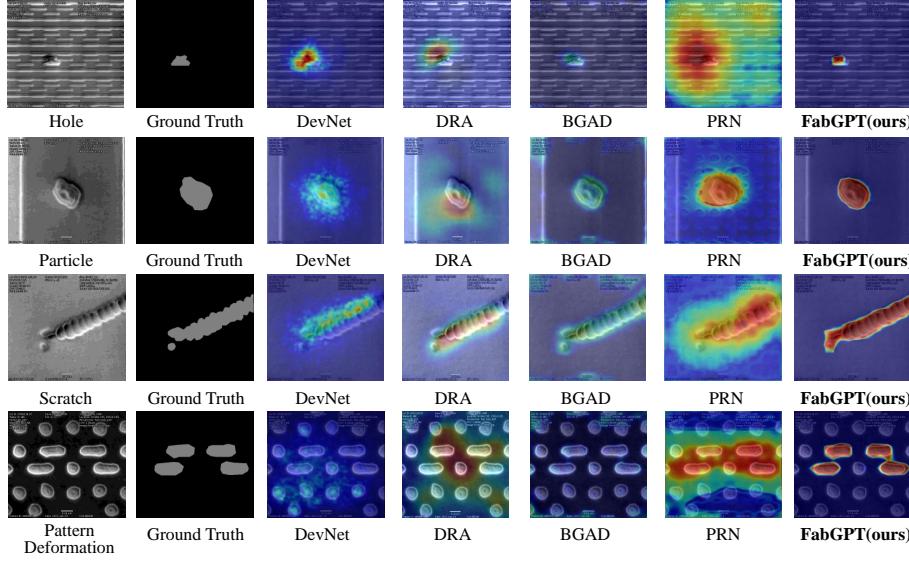| Hole | Ground Truth | DevNet | DRA | BGAD | PRN | **FabGPT(ours)** |
Particle	Ground Truth	DevNet	DRA	BGAD	PRN	**FabGPT(ours)**
Scratch	Ground Truth	DevNet	DRA	BGAD	PRN	**FabGPT(ours)**
Pattern Deformation	Ground Truth	DevNet	DRA	BGAD	PRN	**FabGPT(ours)**

Figure 6: Visualization results of four defect types, compared with non-LMM baseline models.

Comparing heatmaps in Figure 6, we observe that our model focuses more on defect regions than others, while avoiding concentrating too much on normal regions. Moreover, comparing mask images in Figure 7, FabGPT provides more accurate edge segmentation details compared to both Lisa [20] and AnomalyGPT [19].

Q&A Task:

According to Figure 8 and Figure 9, the proposed FabGPT offers detailed descriptions of defect knowledge, providing engineers with more useful information. It should be noted that the questions shown in the figures are not included in our training Corpus-A and Corpus-B.

Specifically, FabGPT is proficient in identifying the defect in SEM images, localizing its position, and discerning specific properties, such as "hole" evident in Figure 8 and "scratch" depicted in Figure 9. In addition, FabGPT can analyze the root causes of wafer defects and propose appropriate solutions from a process perspective, thereby serving as an effective compass for refining the production line.

Furthermore, FabGPT demonstrates competency in providing accurate answers to general IC questions that are not closely related to the input images, depicted in Figure 9, such as explaining the process steps in IC manufacturing, the steps of lithography, and how AI is advancing the development of IC manufacturing. This verifies the effectiveness of our proposed modulation module in resolving modality bias.

Table 4: The ablation results of important components in each stage are recorded using the Pixel-AUC and the Q&A accuracy (%) of defect-related. The best results are highlighted.

Components	Stage	Task		Alleviate Bias
		Pixel-AUC	Defect-Related	
+ Text Mark	Text Input	88.61	83.33	×
+ PM	Enhancement	92.57 (+3.96%)	86.67 (+3.34%)	×
+ Pre-trained Experts	Enhancement	95.61 (+3.04%)	93.33 (+6.66%)	×
+ Q-Former	Q&A	95.61 (+0.00%)	96.67 (+3.34%)	×
+ Corrector	Q&A	95.61 (+0.00%)	96.67 (+0.00%)	✓

4.4 Ablation Studies

In this section, we conduct ablation studies to demonstrate the importance of each component introduced at every stage and the operations within them.

Stage Components:

We study the proposed components in each stage as shown in Table 4. It confirms the importance of each individual component and the best way to structure the sub-stage. It can be seen that the prediction of the PM and pre-trained experts significantly improved defect detection in the modal enhancement stage, and the operations in the modulator played a crucial role in addressing bias issues in the Q&A stage.
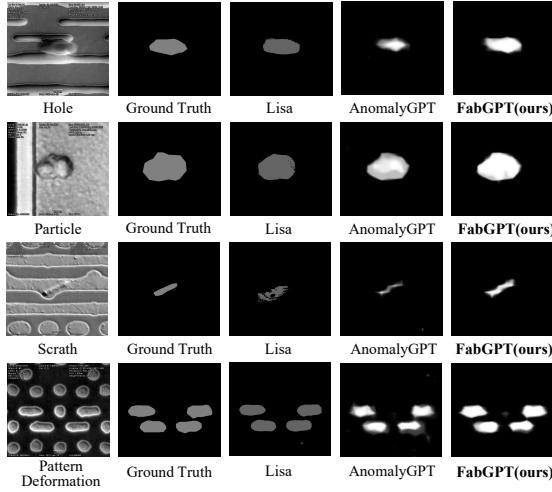
Design for PM Operation:

Figure 7: Visualization results of four defect types, compared with LMM-based baseline models.
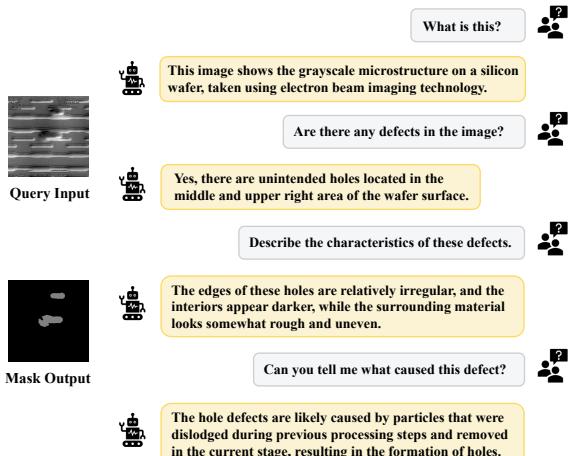

Figure 8: Dialogue example of FabGPT for "hole" type defects.

Table 5: Ablation studies comparing different methods in the PM, with the AUC metric used to evaluate detection accuracy.

Strategy		Image-AUC	Pixel-AUC
-	BiLinear Similarity	85.77	89.33
-	Matrix Multiplication	88.25	92.59
-	Cosine Similarity	91.03	94.61
Linear + ReLU	**Cosine Similarity**	**91.81**	**95.61**

Table 5 shows the advantages of the operation strategy we designed for the PM. It can be seen that compared to matrix multiplication or other operations, using the cosine of the angle to calculate the similarity between two vectors provides a better assessment of their relationship. Moreover, when combined with Linear and ReLU, it achieves a better computational effect.
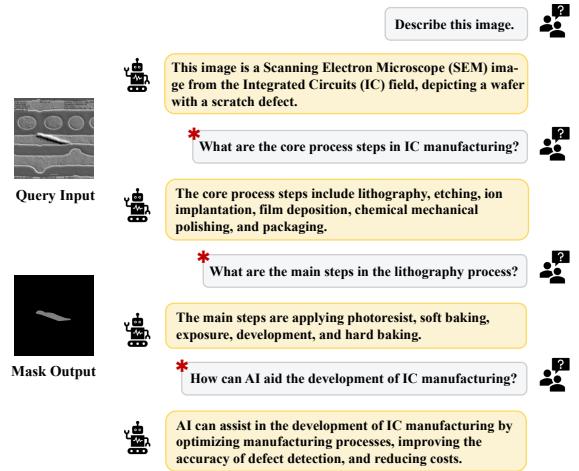

Figure 9: Dialogue example of FabGPT for "scratch" type defects. Questions with "∗" are general IC questions that are not closely related to the input images.

Table 6: Ablation experiments on the embedding schemes of visual prompt instructions to validate the Q&A functionality.

Visual Instruction	Defect-Related	Unrelated
T_{img}	82.50	10.00
$T_{img} + T_{txt}$	85.00	10.00
$T_{img} + T_{txt} + T_{mas}$	95.00	10.00
$T_{vis} + T_{mas}$	**96.67**	12.00
$aT_{vis} + T_{mas}$	**96.67**	**98.00**

Embedding Scheme of Prompt Instructions:

From Table 6, it can be seen that our embedding scheme for prompt instructions showcases optimal effectiveness in mitigating modality bias issues. By aligning and updating the corresponding coefficient "a" for visual tokens, our model effectively discerns the relationship between user queries and visual inputs and mitigates the modality bias issues.

5 CONCLUSIONS

In this paper, we introduce a novel large multimodal language model, FabGPT, for defect knowledge querying in the IC field, including defection, analysis, Q&A, *etc*. It employs three stages to gradually achieve the functionality of defect detection and high-quality dialogue. Enhanced feature tokens aid the model in automatically conducting high-precision detection. Dynamically aligned and corrected tokens fine-tune the LLM, enabling not only attribution analysis of defect regions and correct responses regarding defect type, location, quantity, *etc.*, but also mitigate the modality bias issues within conversations. We validate the effectiveness of FabGPT on the SEM-WaD dataset and 100 questions. Our work provides great convenience for the semiconductor industry and also offers insights for further LMM research.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "Pandagpt: One model to instruction-follow them all," *arXiv preprint arXiv:2305.16355*, 2023.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[4] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[5] 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.

[6] M. D. M. Reddy, M. S. M. Basha, M. M. C. Hari, and M. N. Penchalaiah, "Dall-e: Creating images from text," *UGC Care Group I Journal*, vol. 8, no. 14, pp. 71–75, 2021.

[7] Y. Gu, L. Dong, F. Wei, and M. Huang, "Pre-training to learn in context," *arXiv preprint arXiv:2305.09137*, 2023.

[8] M. Chen, J. Du, R. Pasunuru, T. Mihaylov, S. Iyer, V. Stoyanov, and Z. Kozareva, "Improving in-context few-shot learning via self-supervised training," *arXiv preprint arXiv:2205.01703*, 2022.

[9] J. Wei, L. Hou, A. Lampinen, X. Chen, D. Huang, Y. Tay, X. Chen, Y. Lu, D. Zhou, T. Ma *et al.*, "Symbol tuning improves in-context learning in language models," *arXiv preprint arXiv:2305.08298*, 2023.

[10] M. Quirk and J. Serda, *Semiconductor manufacturing technology*. Prentice Hall Upper Saddle River, NJ, 2001, vol. 1.

[11] S.-K. S. Fan, D.-M. Tsai, F. He, J.-Y. Huang, and C.-H. Jen, "Key parameter identification and defective wafer detection of semiconductor manufacturing processes using image processing techniques," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 4, pp. 544–552, 2019.

[12] T. Lechien, E. Dehaerne, B. Dey, V. Blanco, S. De Gendt, and W. Meert, "Automated semiconductor defect inspection in scanning electron microscope images: a systematic review," *arXiv preprint arXiv:2308.08376*, 2023.

[13] E. G. Seebauer and K. W. Noh, "Trends in semiconductor defect engineering at the nanoscale," *Materials Science and Engineering: R: Reports*, vol. 70, no. 3-6, pp. 151–168, 2010.

[14] G. Pang, C. Ding, C. Shen, and A. v. d. Hengel, "Explainable deep few-shot anomaly detection with deviation networks," *arXiv preprint arXiv:2108.00462*, 2021.

[15] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, "Convolutional neural network for wafer surface defect classification and the detection of unknown defect class," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 2, pp. 163–170, 2019.

[16] C. Ding, G. Pang, and C. Shen, "Catching both gray and black swans: Open-set supervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7388–7398.

[17] H. Zhang, Z. Wu, Z. Wang, Z. Chen, and Y.-G. Jiang, "Prototypical residual networks for anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 281–16 291.

[18] X. Yao, R. Li, J. Zhang, J. Sun, and C. Zhang, "Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 490–24 499.

[19] Z. Gu, B. Zhu, G. Zhu, Y. Chen, M. Tang, and J. Wang, "Anomalygpt: Detecting industrial anomalies using large vision-language models," *arXiv preprint arXiv:2308.15366*, 2023.

[20] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," *arXiv preprint arXiv:2308.00692*, 2023.

[21] M. Zontak and I. Cohen, "Kernel-based detection of defects on semiconductor wafers," in *2009 IEEE international workshop on machine learning for signal processing*. IEEE, 2009, pp. 1–6.

[22] J. L. Gómez-Sirvent, F. L. de la Rosa, R. Sánchez-Reolid, A. Fernández-Caballero, and R. Morales, "Optimal feature selection for defect classification in semiconductor wafers," *IEEE Transactions on Semiconductor Manufacturing*, vol. 35, no. 2, pp. 324–331, 2022.

[23] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[24] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, p. 6, 2023.

[25] S. Alaparthi and M. Mishra, "Bidirectional encoder representations from transformers (bert): A sentiment analysis odyssey," *arXiv preprint arXiv:2007.01127*, 2020.

[26] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 558–567.

[27] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[28] C. Li, W. Liu, R. Guo, X. Yin, K. Jiang, Y. Du, Y. Du, L. Zhu, B. Lai, X. Hu *et al.*, "Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system," *arXiv preprint arXiv:2206.03001*, 2022.

[29] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 180–15 190.

[30] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 603–612.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[32] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[34] ——, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.