OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Data- and Parameter-Efficient Adaptation for Foundation Models and Applications
Author(s)	Zhang, Jiahao
Citation	大阪大学, 2026, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/105827
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Data- and Parameter-Efficient Adaptation for Foundation Models and Applications

Submitted to

Graduate School of Information Science and Technology

The University of Osaka

January, 2026

Jiahao ZHANG

Abstract

Foundation models have revolutionized artificial intelligence. Although they generalize well, achieving near-optimal performance in specialized domains often requires adaptation beyond off-the-shelf prompting, which can be resource-intensive in supervised or domain-specific settings due to annotation and computational costs. To accommodate real-world constraints, this thesis explores data- and parameter-efficient adaptation, establishing a spectrum of adaptation strategies ranging from parameter tuning to input-level optimization and training-free interaction. We present three complementary projects that advance this premise across different levels of model access. First, regarding parameter-efficient fine-tuning for volumetric medical data, we propose *ILD-Slider*, a framework integrating anatomy-guided slice selection with a novel Slice-level 3D Adapter for efficient diagnosis. Second, shifting to input-level optimization for Visual In-Context Learning (VICL), we propose *E-InMeMo*, which introduces learnable perturbations to input prompts to enhance few-shot performance without backbone tuning. Third, investigating training-free interaction for proprietary generative models where gradients are inaccessible, we propose *VASCAR*, a visual-aware self-correction framework that steers black-box models through iterative visual feedback for superior layout generation. Collectively, these studies demonstrate effective strategies for adapting foundation models across varying degrees of parameter access and data availability, offering scalable solutions for specialized applications.

Contents

Abstract i

1 Introduction 1

2 ILD-Slider: Efficient 3D Adaptation for Medical Imaging 5

2.1 Overview 5

2.2 Related Work 7

2.2.1 Medical Image Classification 7

2.2.2 Parameter-Efficient Fine-Tuning (PEFT) 8

2.2.3 Prognostic Prediction for ILD 9

2.3 Method 10

2.3.1 Study Population 10

2.3.2 Overview 10

2.3.3 Position Marker Selector 11

2.3.4 PF-ILD Identification 14

2.4 Experiments 17

2.4.1 Experimental Setup 17

2.4.2 Results of PSM-Based PM Extractor 21

2.4.3 Results on Slider for PF-ILD Identification 22

2.4.4 Domain Shift Analysis 24

2.4.5 Visualization of TPs, TNs, FPs, and FNs 25

2.4.6 Sensitivity to Dataset Size . . . 25
2.4.7 Further Analyses on ILD-Slider . . . 25
2.5 Discussion . . . 31
2.6 Summary . . . 32

3 E-InMeMo: Pixel-Level In-Context Pair Enhancement for VICL 33
3.1 Overview . . . 33
3.2 Related Work . . . 37
3.2.1 In-Context Learning . . . 37
3.2.2 Learnable Prompting . . . 38
3.3 Method . . . 39
3.3.1 Preliminary: MAE-VQGAN . . . 39
3.3.2 Overview of E-InMeMo . . . 40
3.3.3 In-context Learning Retriever . . . 41
3.3.4 Prompt Enhancer . . . 42
3.3.5 Prediction . . . 43
3.3.6 Training . . . 43
3.3.7 Interpretation . . . 44
3.4 Experiments . . . 45
3.4.1 Experimental Setup . . . 45
3.4.2 Comparison with State-of-the-Art . . . 47
3.4.3 Results on Medical Datasets . . . 48
3.4.4 Domain Shifts Analysis . . . 49
3.4.5 Further Analyses on E-InMeMo . . . 50
3.5 Discussion . . . 59
3.6 Summary . . . 60

4 VASCAR: Training-Free Visual Self-Correction for Layout Generation 61
4.1 Overview . . . 61

4.2 Related Work 64
4.2.1 Layout-Specific Generative Models. 64
4.2.2 Layout Generation by LLMs. 65
4.2.3 Self-Correction in LLMs 66
4.3 Method 66
4.3.1 Task Definition 67
4.3.2 Visual-Aware Self-Correction 68
4.3.3 Prompt Strategy for Layout Generator 69
4.3.4 Multi-modal Prompt Optimizer 70
4.4 Experiments 71
4.4.1 Experimental Setup 71
4.4.2 Input-unconstrained Generation 73
4.4.3 Input-constrained Generation 77
4.4.4 Ablation Study 79
4.4.5 Prompt Examples 84
4.5 Summary 92

5 Conclusion **93**

Acknowledgements **96**

Reference **98**

List of Publications **124**

Awards **127**

List of Figures

2.1 Overview of the proposed ILD-Slider framework. A PM extractor and PSM are used to identify the predefined PMs: apical lung, tracheal bifurcation, and upper diaphragm. To address severe class imbalance, the PM extractor is trained on a class-balanced dataset constructed by oversampling the PM classes and undersampling the Non-PM class. Based on the extracted PMs, RSs are selected for subsequent PF-ILD identification. Finally, a lightweight slice-level 3D Adapter (Slider), equipped with tunable adapters in each transformer block, performs the final PF-ILD identification. 11

2.2 Visual examples of the position markers (PMs) and representative slices (RSs). The locations of the PMs and RSs are indicated on the longitudinal CT view, with the corresponding axial CT slices shown alongside (best viewed in color). The PMs include apical lung (upper), tracheal bifurcation (middle), and upper diaphragm (lower). The RSs are shown from top to bottom as 3-RS, 5-RS, and 9-RS. Here, -1 and $+1$ denote the adjacent slices superior and inferior to the current slice, respectively. 15

2.3 (**a**) Scanner mask normalization: We extract the scanner mask from UOH and use it on data from OTMC to mitigate the disparity across the two facilities. (**b**) RGB windowing processing: We perform three different types of organ-specific windowing and concatenate them into an RGB image. “PE” stands for pulmonary embolism. 16

2.4 Statistics on the extracted slices from the PSM-based PM extractor on EfficientNet-b4. 21

2.5 Visual examples under the 5-RS setting of TP, TN, FP, and FN cases from the OTMC and UOH datasets. Each case presents the enhanced lung window image (Enhanced Lung), the RGB windowing-processed image (RGB Image), and the Grad-CAM attention map (Attention Map) corresponding to the predicted class. 26

2.6 Performance comparison of full fine-tuning and Slider in terms of AUROC across different training data fractions. 27

3.1 A schematic comparison between standard visual ICL and E-InMeMo. (**a**) Visual ICL constructs a four-cell grid canvas composed of a query image, an in-context pair, and an empty cell (bottom-right) where the model generates the prediction. This canvas serves as the *prompt*, and the output (marked in the red box) is produced by passing it through a frozen large-scale vision model. (**b**) E-InMeMo enhances this paradigm by introducing a *learnable prompt*, a trainable perturbation designed to adjust the distribution of the in-context pair, thereby improving task guidance and prediction accuracy. 34

3.2 Performance comparison of visual ICL on a foreground segmentation task. Blue boxes indicate in-context pairs, while red boxes represent the predicted label images (query images are unmarked). The quality and similarity of the in-context pair significantly influence the prediction results. In the absence of a learnable prompt, performance is highly dependent on the semantic closeness between the query and in-context images. In contrast, E-InMeMo, with its learnable prompt, produces more stable and accurate predictions across varying input conditions. 35

3.3 Overview of the proposed E-InMeMo framework. The process begins with the In-context Learning Retriever, which selects an in-context pair from the dataset $\mathcal{S}$ for a given query image. A Prompt Enhancer $t_\phi(\cdot)$ then applies learnable perturbations to the in-context pair, producing an enhanced version. These enhanced in-context images, along with the query and an empty cell (bottom-right), are arranged into a four-cell grid canvas. This canvas is passed through a frozen `MAE`, which outputs predicted visual tokens corresponding to the empty cell. For visualization, the predicted tokens are decoded into an image using the decoder of `VQGAN`. During training, a ground-truth canvas, comprising the original in-context pair and the true label for the query, is encoded using a pre-trained `VQGAN` encoder to produce ground-truth tokens. A cross-entropy loss is computed on the empty cell to update only the parameters of the Prompt Enhancer, ϕ. 41

3.4 Qualitative comparisons across baseline methods, prompt-SelF, and our proposed E-InMeMo on two downstream tasks: (**a**) Foreground segmentation and (**b**) Single-object detection. For each task, the top row shows the query image. The subsequent rows (from top to bottom) present results from FMLR (DINOv2), prompt-SelF, **E-InMeMo**, and the ground-truth (GT) label, respectively. E-InMeMo consistently enables visual ICL to capture finer details and demonstrates robustness to mismatches between in-context and query images. Notably, it also appears to mitigate the negative impact of low-quality in-context pairs—an important advantage when the retriever fails to find highly similar samples. 51

3.5 Qualitative results of FMLR (DINOv2) and our proposed E-InMeMo on two medical datasets: (**a**) Kvasir and (**b**) ISIC. For each dataset, the top row shows the query image. The following rows are arranged from top to bottom in the order of FMLR (DINOv2), **E-InMeMo**, and the ground-truth label (GT), respectively. 52

3.6 Intra- and inter-class generalization performance in mIoU. The horizontal and vertical axes are the classes used for prediction and training, respectively. The diagonal elements show intra-class performance. 54
3.7 The performance of E-InMeMo and FMLR (DINOv2) in mIoU of each fold for the size of retrieval set $\mathcal{S}$. We annotate the scores in the figure. 56
3.8 The performance of E-InMeMo in mIoU of each fold for the number of images per class in $\mathcal{S}$. All means to use all images in the training set. 57

4.1 The designers' workflow follows an iterative revision process [1] and heuristics based on guidelines [2]. The conversation highlights the ability of VLMs to evaluate and identify design issues on *rendered* problematic layouts, and provide valuable feedback. This capability is leveraged *appropriately* to improve content-aware layout generation, like the workflow of designers. 62
4.2 An overview of VASCAR. First, *ICL retriever* uses the saliency map of the query image to retrieve several ICL examples from the dataset. The retrieved samples are transformed to rendered images along with the text layout in HTML format. We input all the ICL examples and the query into a frozen VLM to generate the layout in HTML format. The VLM is queried iteratively with optimized prompts by *layout scorer* and *suggester*. After I iterations, VASCAR return the candidate with the highest score. 67
4.3 Visual comparison of baselines and VASCAR with different values of I. 75
4.4 The trends of each metric based on the number of self-correction I. 76
4.5 User study results on PKU test split. 77
4.6 Visualization of the candidate pools under self-correction process with VASCAR. Best viewed in color. 82
4.7 The distribution of generated layout results (Predicted Boxes) (coordinates: left, top, width, and height) and ground truth (GT Boxes) for the unconstrained layout generation task on the PKU and CGL test splits. 85

4.8 A prompt example of **C** $\rightarrow$ **S + P** task. 86
4.9 A prompt example of **C + S**$\rightarrow$**P** task. 87
4.10 A prompt example of **Completion** task. 88
4.11 A prompt example of **Refinement** task. 89
4.12 A prompt example of **Relationship** task. 90
4.13 A prompt example of optimized prompt. 91

List of Tables

2.1 Number of non-PF-ILD and PF-ILD cases from the two facilities. UOH and OTMC denote The University of Osaka Hospital (UOH) and the National Hospital Organization Osaka Toneyama Medical Center (OTMC), respectively. . . 18
2.2 The results of the PSM-based PM extractor with 1-Up-Down accuracy. The best scores for each PM are highlighted in **bold**. 21
2.3 Performance of Slider on the PF-ILD datasets across different metrics for 3-RS, 5-RS, and 9-RS settings. #Params denotes the number of tunable parameters. Rows shaded in light blue indicate our method. The best score for each setting is shown in **bold**. 23
2.4 Cross-domain evaluation of Slider on AUROC. Each row corresponds to the facility used for training, and each column corresponds to the facility used for testing. 24
2.5 The performance of Slider for different scale factors, δ. The row shaded in light blue indicates our default settings. 28
2.6 The performance of Slider with (w/) and without (w/o) RGB windowing processing. The row shaded in light blue indicates our default settings. 28
2.7 The performance of Slider on randomly selected slices (mean $\pm$ std over three runs with different random seeds) and RSs. The row shaded in light blue indicates our method. 29

2.8 The performance of Slider on different kernel shape settings. The row shaded in light blue indicates our default settings. 30
2.9 The performance of single- and multi-layer linear probing on 5-RS. 30

3.1 Performance of the foreground segmentation and single object detection downstream tasks. The best scores in *in-context learning* are highlighted in **bold**. **Seg.** and **Det.** stand for the foreground segmentation and single object detection tasks, respectively. 48
3.2 The performance of E-InMeMo on ISIC and Kvasir datasets with different diseases. 49
3.3 Domain shifts analysis on E-InMeMo. *Pascal* $\rightarrow$ *Pascal* means in-context pairs and query images are both from PASCAL. *COCO* $\rightarrow$ *Pascal* indicates that in-context pairs are from COCO while query images are from PASCAL. 50
3.4 Segmentation performance for different combinations of images in the canvas to which the learnable prompt is applied. `I`, `L`, and `Q` denote the in-context image, in-context label image, and query image, respectively. 53
3.5 The mIoU scores for varying padding sizes of prompt enhancer t_ϕ. The Para. represents the number of tunable parameters. The highest score in each fold is marked in **bold**. 58

4.1 Unconstrained generation results on the PKU and CGL test split. Real Data is the test set with ground truth. † shows our reproduced results based on the RALF split [3]. Best and second-best results are **bold** and with underline, respectively. 73
4.2 Unconstrained generation results on the unannotated test split. Best and second-best results are **bold** and with underline. 74
4.3 Quantitative result of five constrained generation tasks on the PKU and CGL test splits. Best and second-best results are in **bold** and underline, respectively. 78
4.4 Multi-modal analysis for VASCAR on the PKU test split. 79

4.5 Comparison of results across various experimental settings on the PKU test split. “Initial” means without self-correction. “w/o” means without. The blue and red values indicate improved and degraded metrics compared to VASCAR. 80

4.6 Comparison of post-hoc constraint optimization with LACE [4] and our VASCAR on the PKU test split. Blue and red numbers show improved and degraded metrics compared to Initial. 81

4.7 Comparison of results under different settings for ICL example selection on the PKU test split. 83

4.8 Comparison of results across different temperature settings of VLM on the PKU test split. 84

Chapter 1

Introduction

Foundation models, encompassing large-scale vision-language models (VLMs) [5–7] and vision foundation models (VFMs) [8–10], have been widely adopted in natural language processing (NLP) and computer vision. These models demonstrate remarkable capabilities across various downstream tasks, such as classification [11], segmentation [12], and object detection [13]. A central motivation for developing foundation models is to enable broad transfer learning across tasks and domains [14]. Although full fine-tuning can serve as an upper-bound baseline, foundation models are often designed to be adapted without updating all parameters, and full fine-tuning is frequently undesirable in practice and can be infeasible for proprietary or closed models. Specifically, updating all parameters is parameter-inefficient, since it typically requires maintaining a separate set of model weights for each task [15]. Furthermore, full fine-tuning often demands massive high-quality datasets and substantial computational resources. Consequently, for widespread real-world applications, full fine-tuning frequently becomes impractical when accounting for annotation costs, the scale of trainable parameters, and deployment constraints such as latency or memory limits.

To address these constraints, this thesis focuses on data- and parameter-efficient adaptation methods for foundation models and their effective applications. Broadly speaking, the literature on efficient adaptation is dominated by two paradigms: (1) parameter-efficient fine-tuning (PEFT), which adapts a model by updating only a small set of parameters, and (2) prompting-

based steering, including in-context learning (ICL), which conditions a frozen model on task instructions and in-context examples without updating its weights.

Parameter-Efficient Fine-Tuning (PEFT). Given the prohibitive computational cost of full fine-tuning, efficient alternatives have been extensively explored in the NLP field, particularly for large-scale language models (LLMs) [16]. These approaches are known as Parameter-Efficient Fine-Tuning (PEFT). Representative PEFT methods, such as adapters [15, 17, 18] and learned prompt vectors [19, 20], achieve performance comparable to full fine-tuning with significantly reduced overhead in terms of parameters and computational resources. Inspired by their success in NLP, researchers have extended these PEFT strategies to the computer vision field [21–24] and vision-language models [8, 25–27].

However, while the adaptation of foundation models to medical imaging has become a major research area, efficiently bridging the gap between 2D VFMs and 3D volumetric data (e.g., capturing inter-slice dependencies in CT scans) remains a critical challenge [28]. In these scenarios, task-specific high-quality annotations are often limited compared to natural images, and effectively processing volumetric information without incurring the high computational cost of full 3D models presents unique difficulties for adaptation.

Steering frozen models via prompting. Since foundation models possess intrinsic general capabilities, steering them efficiently without parameter updates has become a key research focus [29]. Moreover, with the emergence of proprietary generative models, such as GPTs [5, 6, 30] and Gemini [7], gradients are often inaccessible for any form of tuning. Fortunately, In-Context Learning (ICL) offers a paradigm for steering such frozen models, with the potential to become the mainstream approach for Model-as-a-Service (MaaS) applications [31]. Pioneered by GPT-3 [30] in NLP, ICL enables foundation models to perform novel tasks on unseen data by leveraging pre-defined prompts (e.g., input-output text pairs) without additional training [32–39]. Beyond NLP, Visual In-Context Learning (VICL) [10, 31, 40–42] has also emerged. Analogous to NLP, pre-defined prompts are transferred from text to images (e.g., input-output image pairs, also known as in-context pairs) to perform various downstream tasks, often formulated as inpainting problems.

Unlike ICL in NLP, current VICL paradigms are typically highly sensitive to the quality of prompts; visual discrepancies (such as variations in appearance, viewpoint, and object size) between the prompt and query often lead to degraded performance [31, 43]. Furthermore, considering the frozen backbone in VICL and the inaccessibility of gradients or internal architectures in proprietary generative models, exploring how to effectively steer these models via input-level optimization or training-free mechanisms is crucial.

Therefore, the versatility of efficient adaptation requires tailored strategies for different levels of model access and data availability, spanning from parameter tuning to input-level optimization and training-free interaction. In this thesis, we explore efficient adaptation through this spectrum. Our approach follows a step-by-step progression: First, we address the adaptation of VFMs for the medical field from a parameter-tuning perspective. Next, we explore input-level visual prompt tuning for VICL, enhancing the performance of frozen foundation models. Finally, we investigate a training-free mechanism for steering proprietary generative models.

In our first work presented in Chapter 2, we revisit PEFT methods and introduce *ILD-Slider*, a framework that adapts 2D VFMs [9] to complex 3D diagnostic tasks, specifically the challenging diagnosis of Progressive Fibrosing Interstitial Lung Disease (PF-ILD). While standard image-level adapters fail to capture inter-slice dependencies, full 3D models are computationally expensive, data-inefficient, and prone to overfitting. To bridge this gap, ILD-Slider incorporates a novel Slice-level 3D Adapter (Slider) into the backbone, enabling the model to learn volumetric representations with minimal trainable parameters (approximately 4.1% of total parameters). Furthermore, by introducing an anatomy-guided slice selection strategy, ILD-Slider systematically reduces the number of input slices required for diagnosis. We demonstrate that ILD-Slider outperforms fully fine-tuned baselines in identifying PF-ILD. This study exemplifies how carefully designed PEFT methods can endow 2D VFMs with 3D capabilities for specialized medical domains.

Next, in Chapter 3, we explore input-level optimization to enhance VICL, shifting our focus from modifying intrinsic model parameters to optimizing input prompts. Existing VICL methods rely heavily on retrieving similar in-context pairs via off-the-shelf retrievers but often fail

when there are mismatches in appearance, viewpoint, or object size between the query and the in-context pair. To address this, we propose *E-InMeMo*, a framework that introduces learnable perturbations to the input in-context pairs. By optimizing the lightweight visual prompting to bridge these visual discrepancies, E-InMeMo significantly boosts the performance of frozen VICL models on segmentation and detection tasks. Furthermore, it demonstrates robustness to domain shifts and applicability to medical domains. This project illustrates that optimizing the input space via learnable prompting serves as a powerful, data- and parameter-efficient alternative to model tuning.

Finally, in Chapter 4, we investigate the potential of efficient adaptation from a training-free perspective using proprietary generative models (e.g., GPT-4o [6] and Gemini [7]) for complex content-aware layout generation. For proprietary VLMs where neither parameters nor gradients are accessible, traditional fine-tuning or prompt optimization is infeasible. To overcome this, we propose *VASCAR*, a visual-aware self-correction framework for progressive optimization. Inspired by the iterative workflow of human designers and the inherent evaluation capabilities of VLMs, VASCAR enables models to iteratively refine their generated layouts based on visual feedback from rendered images and heuristic guidelines, comprising evaluation scores and specific suggestions. This work achieves state-of-the-art generation quality without any additional training or tuning, highlighting how multi-modal prompting strategies can unlock the latent generative capabilities of proprietary foundation models.

The remainder of this thesis is organized as follows. Chapter 2 details our PEFT method for a real-world disease diagnosis application by devising slice-level adapters and an anatomy-guided slice selection strategy. Chapter 3 introduces the optimization of learnable visual prompting for VICL, steering frozen models effectively from the input level. Chapter 4 presents a training-free method for prompting proprietary generative models. Finally, Chapter 5 concludes the thesis.

Chapter 2

ILD-Slider: Efficient 3D Adaptation for Medical Imaging

2.1 Overview

Interstitial Lung Disease (ILD) is a chronic condition characterized by lung infiltration and fibrosis, ultimately leading to respiratory failure. Early diagnosis is crucial for improving patient survival; however, it remains challenging due to the heterogeneous clinical presentations of ILD [44]. Although lung biopsy is considered the gold standard for diagnosis, it is not always feasible because of patient-related factors such as contraindications, patient preference, or an inability to undergo the procedure. Moreover, CT examinations over a defined observation period are often required to assess fibrotic progression, inevitably delaying both diagnosis and treatment initiation.

Diagnosing ILD demands the expertise of specialized pulmonologists, radiologists, and pathologists, which may not always be readily available. Accurate diagnosis is difficult, and predicting disease progression is even more challenging due to variability among patients. One subtype, Progressive Fibrosing Interstitial Lung Disease (PF-ILD), carries a poor prognosis, with a median survival of only 3–5 years after diagnosis [45–47]. Recently, the antifibrotic

drugs nintedanib and pirfenidone have been shown to slow fibrosis and extend survival [48,49], making timely and accurate PF-ILD identification essential for initiating treatment. This underscores the need for reliable early diagnostic methods.

Deep learning has led to significant advances across multiple domains, including computer-aided diagnosis [50–52], autonomous driving [53], and smart cities [54,55]. In computer vision, the emergence of vision foundation models [8, 9, 12] has enabled high performance in large-scale, complex tasks. Medical image analysis has similarly benefited, achieving remarkable diagnostic accuracy using image data alone [56, 57] or in combination with textual information [58]. These advances have inspired the development of frameworks to support PF-ILD identification. Typically, identification relies on high-resolution CT (HRCT) scans, which provide 3D volumetric data. However, applying 2D foundation models to 3D PF-ILD identification poses two main challenges: (1) large annotated datasets are scarce for fine-tuning, and (2) full 3D models involve a vast number of trainable parameters, demanding substantial computational resources.

These challenges highlight the urgent need for data-efficient and lightweight diagnostic frameworks. Parameter-Efficient Fine-Tuning (PEFT) has gained attention in natural language processing (NLP) for adapting large models to specific downstream tasks with minimal parameter updates. Methods such as task-specific adapters [15, 59, 60] and prompt tuning [61–64] introduce small learnable modules or tokens, substantially reducing the number of trainable parameters. PEFT approaches have also been explored in computer vision, including prefix tuning [25, 26], adapters [65, 66], and prompt tuning [21, 43, 67, 68].

Building on the success of PEFT methods [15,65], in this chapter, we explore efficient adaptation from the parameter-tuning perspective. We propose ILD-Slider, a parameter-efficient, lightweight framework for PF-ILD identification. ILD-Slider integrates a slice-level 3D Adapter (Slider) with a peak slice mining (PSM)-based position marker (PM) extractor to identify representative slices (RSs). At the case level, we define anatomy-based PMs to guide RS selection, as processing all slices is computationally prohibitive. PMs are manually defined as distinct classes and used to train a multi-class classification model. Our PSM-based strategy then se-

lects key slices based on PM indices, from which three, five, or nine RSs are extracted. Finally, these RSs are used to train the Slider for efficient and accurate PF-ILD identification.

The contributions are as follows:

- We introduce a data-efficient approach that employs anatomy-based PMs to guide the selection of RSs for PF-ILD identification. A PM extractor, trained with a standard multi-class classification model, achieves high extraction accuracy despite severe class imbalance. This enables the use of a small set of RSs instead of full CT volumes, thereby improving diagnostic efficiency and significantly reducing computational cost.
- We design a parameter-efficient framework that integrates a PSM-based strategy to automatically select RSs from PM indices. Using datasets from two medical facilities, ILD-Slider achieves an AUPRC of 0.790 and an AUROC of 0.847 with only five extracted RSs. We also propose an effective window level/width processing for PF-ILD identification, validate its effectiveness, and analyze domain shifts on different facilities, demonstrating the robustness of our Slider.
- We demonstrate the feasibility of using non-contiguous slices for PF-ILD identification. This is particularly valuable for real-world and public datasets, where contiguous slices are often unavailable, underscoring the practicality and generalizability of ILD-Slider.

2.2 Related Work

2.2.1 Medical Image Classification

Recently, significant progress has been made in significant progress in applying convolutional neural networks (CNNs) to medical image classification [69], greatly facilitated by the availability of open-source 2D pre-trained models for related medical tasks has greatly facilitated these advancements. Such models, typically pre-trained on large-scale natural image datasets (e.g., ImageNet [70]), can be fine-tuned for various downstream medical applications. Although medical datasets are generally more difficult to obtain than natural image datasets, pre-trained

weights have proven to be highly beneficial in medical imaging. Li et al. [71] proposed a shallow convolutional architecture for classifying ILD in lung image patches. Sakamoto et al. [72] introduced a novel cooperative deep learning method for classifying diverse pathology-based medical images. Nithiyaraj et al. [73] developed a CT slice classification model to assist radiologists in selecting diagnostically valuable slices. Due to the limited availability and imbalance of medical data, weakly supervised, unsupervised, and self-supervised learning approaches have emerged as promising research directions. For example, Li et al. [74] proposed a multi-scale convolutional model that utilizes a shared set of convolutional kernels to extract features with different receptive fields, achieving strong performance in comparison to medical image classification benchmarks.

Inspired by these works, we adopt a conventional CNN model to extract representative slices from each case and address the extreme class imbalance through both oversampling and downsampling. Furthermore, we demonstrate that the proposed method can effectively select a small set of informative slices for PF-ILD identification.

2.2.2 Parameter-Efficient Fine-Tuning (PEFT)

In NLP, prompting is a widely used technique for guiding large language models (LLMs) to better adapt to downstream tasks [16]. For instance, in-context learning (ICL) [30, 33, 37, 39, 75] and visual ICL [10, 41–43, 76] demonstrates remarkable generalization across diverse tasks via carefully designed prompts. However, fully fine-tuning large-scale models is computationally expensive due to the vast number of parameters involved. To address this issue, Parameter-Efficient Fine-Tuning (PEFT) methods have been proposed. These approaches optimize only a small subset of the model's parameters or introduce additional lightweight modules, such as adapters [15, 17] and prompt tuning [19, 20], achieving performance comparable to full fine-tuning while substantially reducing computational cost.

Building on the success of PEFT in NLP, recent research has extended these techniques to computer vision [21, 67] and vision-language modeling [8, 25, 26]. In these domains, PEFT

typically involves either partial fine-tuning of the backbone model or the insertion of learnable prompts into image inputs. For example, Pan et al. [65] enhanced a traditional adapter [15] with a 3D convolutional layer, enabling the Vision Transformer (ViT) [77] to effectively carry out video action recognition tasks. This method updates only a small fraction of the model's parameters, making adapters an attractive solution for computationally constrained medical applications such as CT slice analysis.

Inspired by these developments, we explore adapter-based PEFT methods to improve performance in medical imaging tasks involving CT scans. Unlike standard image-level adapters that operate on each slice independently, we design a slice-level 3D adapter that inserts a lightweight 3D convolution layer within the adapter to aggregate inter-slice context across the input slices, enabling volumetric feature adaptation with a small trainable budget.

2.2.3 Prognostic Prediction for ILD

Furukawa et al. [78] developed a classification approach for idiopathic pulmonary fibrosis (IPF), a major subtype of PF-ILD with a poor prognosis. They employed a semantic segmentation model to delineate fibrotic regions in the lungs and compared the results with non-invasive measures using a diagnostic algorithm for IPF. Ryerson et al. [79] introduced the GAP model to predict mortality risk in chronic ILD. The ILD-GAP model demonstrated robust performance in estimating mortality rates across major subtypes of chronic ILD and at all disease stages. Walsh et al. [80] evaluated the performance of a widely used deep learning model for diagnosing fibrotic diseases—the systematic Objective Fibrotic Imaging Analysis Algorithm (SOFIA)—in predicting usual interstitial pneumonia and found that it outperformed radiologists in identifying progressive fibrotic lung disease.

To the best of our knowledge, only a few published studies have explored machine learning-based approaches for PF-ILD identification. We are the first to propose the use of representative slices for PF-ILD identification and demonstrate promising results.

2.3 Method

2.3.1 Study Population

We collected chest CT scans from patients diagnosed with ILD at The University of Osaka Hospital (UOH) and the National Hospital Organization Osaka Toneyama Medical Center (OTMC). In this study, we utilized a dataset that was previously established in our prior work [81]. Patient inclusion and the definition of PF-ILD followed the same criteria as described therein, namely $\geq$10% fibrosis on HRCT and a relative decline of either $\geq$ 10% or >5% to <10% in forced vital capacity (FVC), accompanied by clinical deterioration or radiological progression during overlapping two-year windows, with baseline $\geq$ 45%. Initially, we identified 1126 ILD cases (UOH, 462; OTMC, 664), which was reduced to 1049 after excluding cases with inadequate CT image quality or incomplete lung coverage (UOH, 28; OTMC, 49). Of these, we further excluded cases missing labels that were required to ascertain PF-ILD status (UOH, 31; OTMC, 405), leaving 613 cases for PF-ILD identification.

2.3.2 Overview

Let $\mathcal{S} = \{x_1, x_2, \ldots, x_N\}$ denote the dataset used for PF-ILD identification, consisting of N cases. Each case is represented as $x_i = \{x_i^1, x_i^2, \ldots, x_i^M\}$, where M denotes the number of CT slices. We employ a classification model S_θ, together with a PSM module to extract PMs. RSs are then selected based on the detected PMs and subsequently fed into the proposed diagnostic framework.

As illustrated in Figure 2.1, we define three types of PMs: apical lung, tracheal bifurcation, and upper diaphragm. A CNN-based PM extractor is trained to identify these PMs, while the PSM module is applied during inference to select the most relevant slices. To address the severe inter-class imbalance during training, we adopt an oversampling strategy for the PM classes and downsample the non-PM class to balance the dataset. After PM extraction, the RSs corresponding to these PMs are selected and input into a vision foundation model (DINOv2 [9]) with a frozen backbone. A slice-level 3D Adapter (Slider) equipped with trainable adapters is

inserted into each transformer block, enabling parameter-efficient fine-tuning for downstream PF-ILD identification.

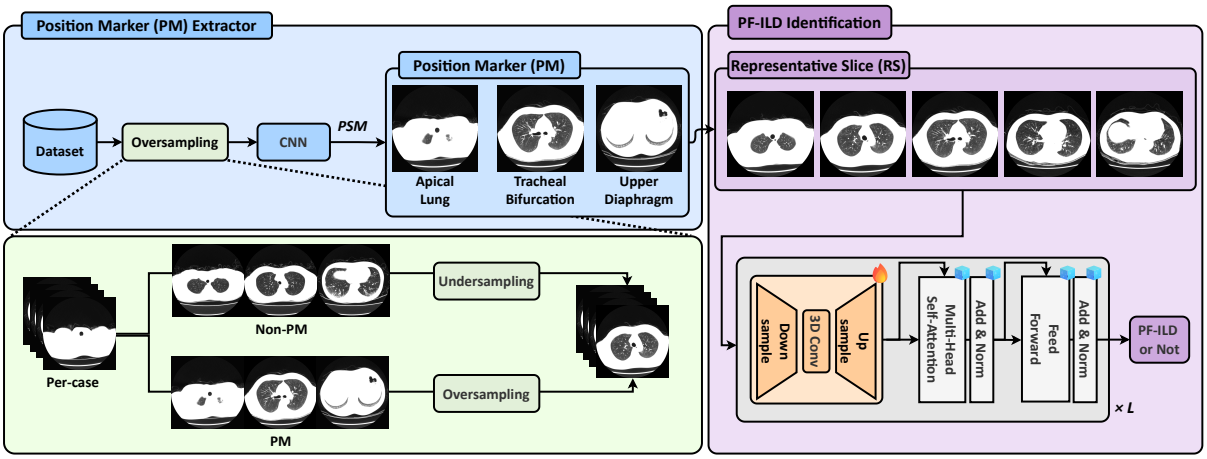


Figure 2.1: Overview of the proposed ILD-Slider framework. A PM extractor and PSM are used to identify the predefined PMs: apical lung, tracheal bifurcation, and upper diaphragm. To address severe class imbalance, the PM extractor is trained on a class-balanced dataset constructed by oversampling the PM classes and undersampling the Non-PM class. Based on the extracted PMs, RSs are selected for subsequent PF-ILD identification. Finally, a lightweight slice-level 3D Adapter (Slider), equipped with tunable adapters in each transformer block, performs the final PF-ILD identification.

2.3.3 Position Marker Selector

PM Extractor Training. In our dataset, the CT scans have a default slice interval of 4 mm, resulting in approximately 75 slices per case. We define and label three PMs to represent distinct anatomical landmarks in the lungs:

The apical lung (upper) PM. The slice where aerated lung parenchyma first appears in either hemithorax is labeled as the upper PM. During the operation, (i) the air in the trachea is ignored, (ii) when both the current slice and the next caudal slice show early parenchyma, the more superior slice is selected, and (iii) at asymmetric onset, the earlier-appearing hemithorax is

used.

The tracheal bifurcation (middle) PM. The slice where the trachea first begins to bifurcate into the right and left main bronchi. A slice is considered the middle PM if all the following criteria hold: (i) when the next caudal slice displays unequivocal bifurcation but the previous superior slice does not, the current slice is selected as the middle PM (the first-bifurcation level), and (ii) two distinct bronchial lumens are present and separated by a soft-tissue wall (no longer a single circular tracheal lumen).

The upper diaphragm (lower) PM. The lower PM is defined as the axial slice located three to four slices superior to the level where aerated lung parenchyma completely disappears bilaterally. We determine the disappearance level using the following operational criteria: (i) the next caudal slice shows no aerated lung parenchyma in both hemithoraces, whereas the previous superior slice still shows aerated parenchyma; (ii) if one lung disappears earlier, we continue scrolling caudally until both lungs are absent; (iii) we ignore any low-attenuation regions that are not aerated lung parenchyma.

The labeling procedure is rule-based and reproducible without requiring radiological expertise. For quality assurance, an experienced physician reviewed and approved the final set of annotations, confirming the anatomical validity of the definitions.

To train a PM extractor, we construct a classification dataset S_{PM}, in which each slice is assigned a PM class label. Since only one slice per case belongs to each PM class, the dataset exhibits severe imbalance between PM and non-PM categories. To address this issue, we apply a combination of oversampling and downsampling: specifically, non-PM slices are randomly downsampled by 75%, while PM slices are oversampled to match the number of non-PM samples. Furthermore, we adopt a weighted loss function to penalize false positives and further mitigate the effects of class imbalance.

Peak Slice Mining (PSM). After training, we apply a case-level PSM function, r, to select one slice per PM class for each case. Given a case, $x_n \in \mathcal{S}$, with M slices, the PM extractor, S_θ, outputs a confidence matrix, $\mathbf{C}_n \in \mathbb{R}^{M \times C}$, where C is the number of PM classes.

To obtain a normalized confidence distribution over slices for each class, we apply the soft-

max function, $\sigma(\cdot)$, along the slice dimension M, independently for each class. For every PM class c, this ensures that the probabilities across all M slices in the case sum to 1. Formally, we can express this as follows:

$$\sigma(\mathbf{C}_n)_{m,c} = \frac{\exp(C_{n,m,c})}{\sum_{m'=1}^{M} \exp(C_{n,m',c})}, \tag{2.1}$$

$$r_c = \underset{m \in \{1,\ldots,M\}}{\arg\max}\ \sigma(\mathbf{C}_n)_{m,c}, \quad \forall c \in \{1,\ldots,C\}. \tag{2.2}$$

where $\sigma(\mathbf{C}_n)_{m,c}$ denotes the normalized probability that the m-th slice in case x_n corresponds to PM class c, and $C_{n,m,c}$ denotes the scalar element of the confidence matrix at index (m, c). For each $c \in \{1,\ldots,C\}$, we select the slice index r_c with the highest probability as the representative PM slice. Formally, the PSM function r maps the confidence matrix $\mathbf{C}_n$ to the set of selected indices:

$$\mathcal{R}_n = r(\mathbf{C}_n) = \{r_1, r_2, \ldots, r_C\}. \tag{2.3}$$

The set $\mathcal{R}_n$ thus contains the index of the peak slice for each PM class. The corresponding slices are subsequently input into the downstream PF-ILD identification task.

Representative Slice (RS) Extraction. During preprocessing, DICOM files are converted into PNG format, and slices within each case are renamed sequentially from the apical lung to the diaphragm. This conversion facilitates RS extraction based on the predefined PMs. We adopt three RS extraction strategies, using three, five, or nine RSs, as shown in Figure 2.2. For the 3-RS strategy, the upper RS is defined as the slice located at one-third of the distance between the apical lung PM and the tracheal bifurcation PM; the middle RS is the tracheal bifurcation PM itself; and the lower RS is the slice located at one-third of the distance between the tracheal bifurcation PM and the upper diaphragm PM. For the 5-RS strategy, we select slices at one-third and two-thirds of the interval between the apical lung PM and the tracheal bifurcation PM, the tracheal bifurcation PM itself, and two-thirds and one-third of the interval between the tracheal bifurcation PM and the upper diaphragm PM. For the 9-RS strategy, we extend the 3-RS approach by including the immediate neighboring slices before and after each of the three selected RSs, resulting in nine slices in total.

By anchoring RS extraction to PMs, we eliminate the need to account for inter-patient variations in lung size and morphology. This approach allows the model to focus on consistent, anatomically defined features, thereby standardizing RS selection across all cases.

2.3.4 PF-ILD Identification

Preliminaries. Adapters [15] were originally introduced as a PEFT technique in NLP. An adapter module typically consists of a down-projection linear layer followed by an up-projection linear layer. Given input features, $\mathcal{X} \in \mathbb{R}^{L\times d}$, at the i-th layer, the adapter transformation can be formulated as follows:

$$\texttt{Adapter}(\mathcal{X}) = \mathcal{X} + f(\mathcal{X}\mathbf{W}_{\text{down}})\mathbf{W}_{\text{up}}, \tag{2.4}$$

where $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times p}$ is the down-projection weight matrix, $\mathbf{W}_{\text{up}} \in \mathbb{R}^{p\times d}$ is the up-projection, and $f(\cdot)$ denotes a non-linear activation function. The bottleneck dimension p is typically defined as $p = \frac{d}{\delta}$, where δ is the reduction factor.

Adapters have shown strong performance in NLP tasks, as they only introduce a small number of task-specific trainable parameters while keeping the backbone model frozen. This not only reduces memory usage and computational cost but also mitigates catastrophic forgetting, a common issue with full fine-tuning [15].

Slice-Level 3D Adapter (Slider). A conventional adapter is limited to 2D feature representations and only performs spatial modeling across tokens. However, CT scans are inherently 3D, and in PF-ILD identification, the spatial arrangement of slices along the axial dimension (superior to inferior) encodes critical anatomical and pathological progression. Inspired by the ST-Adapter [65], we extend the standard adapter by introducing a slice-level 3D Adapter (Slider) to efficiently capture inter-slice dependencies. Similarly to the original adapter, our Slider includes a down-projection layer, a non-linear activation layer, and an up-projection layer. Between these layers, we insert a depth-wise separable 3D convolution layer to perform slice-level reasoning.

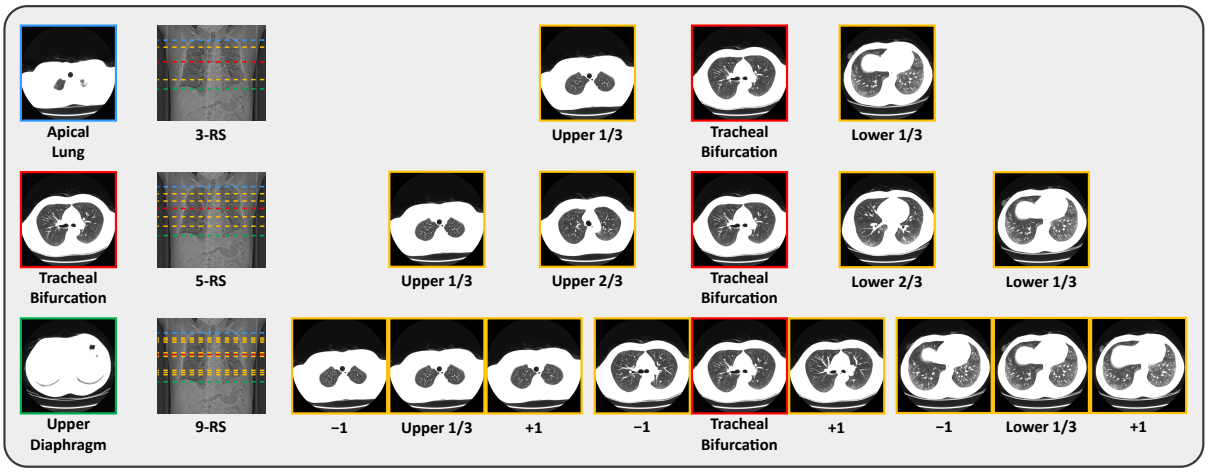


Figure 2.2: Visual examples of the position markers (PMs) and representative slices (RSs). The locations of the PMs and RSs are indicated on the longitudinal CT view, with the corresponding axial CT slices shown alongside (best viewed in color). The PMs include apical lung (upper), tracheal bifurcation (middle), and upper diaphragm (lower). The RSs are shown from top to bottom as 3-RS, 5-RS, and 9-RS. Here, -1 and $+1$ denote the adjacent slices superior and inferior to the current slice, respectively.

Formally, the Slider transformation is defined as follows:

$$\texttt{3D-Adapter}(\mathcal{Z}) = \mathcal{Z} + f(\texttt{3DConv}(\mathcal{Z}\mathbf{W}_{\text{down}}))\mathbf{W}_{\text{up}}, \tag{2.5}$$

where $\mathcal{Z} \in \mathbb{R}^{T\times L\times d}$ denotes the stacked patch features from T CT slices, $f(\cdot)$ is a non-linear activation function, and $\texttt{3DConv}$ is a depth-wise 3D convolution operating across the slice dimension T and the spatial patch grid. As in the conventional adapter, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times p}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{p\times d}$ are the down-projection and up-projection weight matrices, respectively. The bottleneck dimension p is defined as $p = \frac{d}{\delta}$, where δ is the reduction factor. Before applying $\texttt{3DConv}$, the down-projected features are reshaped from sequence format to volumetric form, i.e., $\mathcal{Z}' \in \mathbb{R}^{T\times L\times p} \rightarrow \mathcal{Z}'' \in \mathbb{R}^{T\times H\times W\times p}$, where H and W denote the patch grid height and width, respectively. This reshaping enables the 3D convolution to jointly model spatial and slice-level contexts.

By inserting this adapter module into each transformer block of a frozen ViT backbone,

the model can extract richer 3D representations without modifying the pre-trained weights. In PF-ILD identification, this allows the model to capture disease-related structural patterns that evolve across lung slices, thereby enhancing PF-ILD identification.

Preprocessing. (1) Scanner mask normalization. In this study, we utilize datasets collected from two different medical facilities. Due to variations in diagnostic equipment, such as differences in CT scanner models (see Figure 2.3a), we observe inconsistencies in non-anatomical regions across scans. To address this issue, we apply a unified scanner mask to both the OTMC and UOH datasets. Specifically, we generate a binary scanner mask from the UOH dataset and subsequently apply it to the OTMC dataset. The UOH dataset is also processed using the same standardized mask to ensure consistency across both datasets.

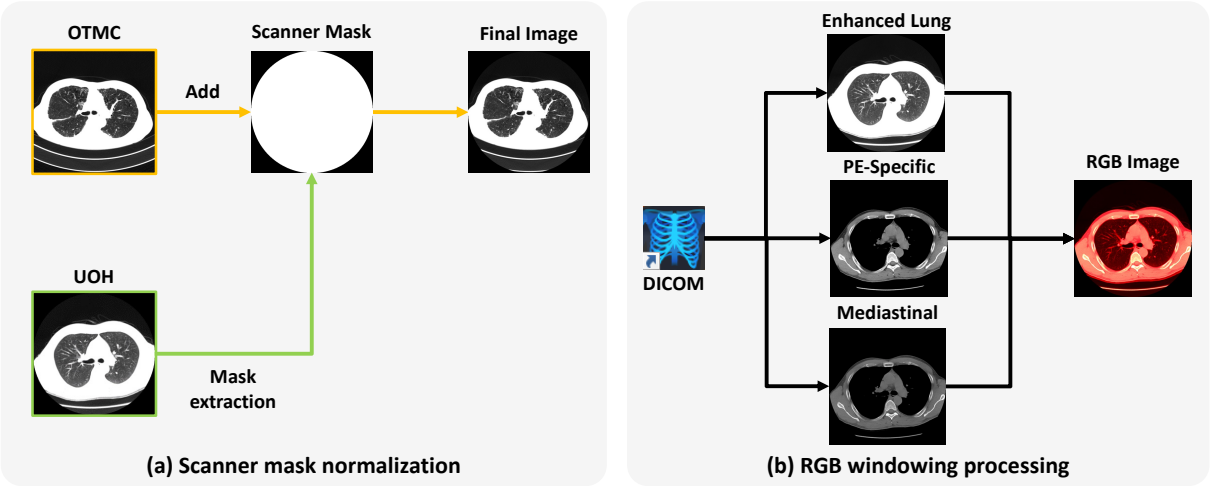


Figure 2.3: (**a**) Scanner mask normalization: We extract the scanner mask from UOH and use it on data from OTMC to mitigate the disparity across the two facilities. (**b**) RGB windowing processing: We perform three different types of organ-specific windowing and concatenate them into an RGB image. "PE" stands for pulmonary embolism.

(2) RGB windowing processing. To enhance visual features that are relevant to PF-ILD, we perform a multi-window image fusion process, starting with the original DICOM scans (see Figure 2.3b). Windowing is performed in Hounsfield units (HU), and each DICOM image is converted into three separate single-channel window images:

- Enhanced lung window: Level = -700 HU; width = 700 HU;
- Pulmonary embolism (PE)-specific window: Level = 100 HU; width = 700 HU;
- Mediastinal window: Level = 40 HU; width = 400 HU.

Each windowed image is resized to 518×518 and then stacked to form a three-channel image of size $518 \times 518 \times 3$, analogous to an RGB image, for compatibility with the DINOv2 pre-trained backbone. This transformation enhances the visibility of fibrotic regions in the lung fields within the PNG outputs, thereby improving the model's ability to accurately identify PF-ILD cases.

2.4 Experiments

2.4.1 Experimental Setup

Datasets. We performed a basic screening process on our datasets, confirming that each patient contributed only one case and ensuring that there was no data leakage between the training and test sets. After screening, 434 cases from the UOH and 615 cases from the OTMC remained, totaling 1049 CT scans. Among these, 403 cases from the UOH and the 210 cases from OTMC were retained with PF-ILD labels, as shown in Table 2.1.

PM extractor dataset: We randomly select 200 cases from each facility (400 cases in total) from the 1049 available cases, regardless of PF-ILD status. All images are converted from the original DICOM format to 8-bit PNG format based on HU, using a Level of -700 HU and a Width of 700 HU, followed by scanner mask normalization. The dataset is split into 70%, 15%, and 15% for training, validation, and testing, respectively.

PF-ILD identification dataset: We use all available cases with PF-ILD labels for identification, totaling 613 cases (403 from the UOH and 210 from the OTMC). These cases undergo scanner mask normalization and RGB windowing processing to better highlight fibrotic regions based on clinical knowledge. The dataset is also divided into 70%, 15%, and 15% for training,

validation, and testing, respectively.

Table 2.1: Number of non-PF-ILD and PF-ILD cases from the two facilities. UOH and OTMC denote The University of Osaka Hospital (UOH) and the National Hospital Organization Osaka Toneyama Medical Center (OTMC), respectively.

Facility	Non-PF-ILD	PF-ILD	Total
UOH	236	167	403
OTMC	130	80	210

Implementation details. All PM extractor and Slider modules are implemented in the PyTorch framework (version 1.13.1) and trained on servers equipped with two NVIDIA RTX 6000 Ada Generation GPUs. For the PM extractor, we use EfficientNet-b4 [82] as the classification backbone. Input images are resized to 256×256 and normalized to the intensity range $[0, 1]$. Models are trained for 20 epochs with a batch size of 256 using weighted cross-entropy loss and the Adam optimizer [83], with an initial learning rate of 1×10^{-3}. Data augmentation is applied during training. For the Slider, we employ DINOv2 [9] as the vision foundation model backbone, with the adapter dimension d set to 192, corresponding to a reduction factor $\delta = 4$. Input images are resized to 518×518. Models are trained for 80 epochs with a batch size of 8 using class-weighted cross-entropy loss (weight ratio 1:1.5), scanner mask normalization, and RGB windowing processing. Optimization is performed using the Adam optimizer, with an initial learning rate of 2×10^{-4}, a cosine annealing learning rate scheduler, and a dropout rate of 0.3.

Evaluation metrics. We adopt several evaluation metrics as shown below.

1-Up-Down Accuracy. To evaluate the performance of the PM extractor, we adopt a relaxed evaluation criterion called 1-Up-Down Accuracy. In clinical practice, the slice immediately above or below the ground-truth PM often contains similar anatomical features. Therefore, we

consider a prediction correct if the predicted slice index, $\hat{m}$, satisfies the following inequality:

$$|\hat{m} - m^*| \leq 1, \tag{2.6}$$

where m^* denotes the ground-truth PM slice index. The 1-Up-Down Accuracy is then defined as follows:

$$\texttt{1-Up-Down Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left(|\hat{m}_i - m_i^*| \leq 1 \right), \tag{2.7}$$

where N is the number of evaluated cases and $\mathbb{1}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

For the PF-ILD identification task, we adopt several metrics for evaluation:

AUROC. The Area Under the Receiver Operating Characteristic Curve (AUROC) is widely used for binary classification and measures the model's ability to distinguish between healthy and diseased samples across various classification thresholds. Given predicted scores, $\hat{y}_i$, and true labels, $y_i \in \{0, 1\}$, for $i = 1, \ldots, N$, the AUROC is defined as follows:

$$\texttt{AUROC} = \frac{1}{N_+ N_-} \sum_{i:y_i=1} \sum_{j:y_j=0} \mathbb{1} \left(\hat{y}_i > \hat{y}_j \right), \tag{2.8}$$

where N_+ and N_- are the number of healthy and diseased samples, respectively.

The metrics below are calculated from the following components of the confusion matrix at a specific threshold of 0.5: True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs).

Accuracy (Acc.) measures the proportion of all samples that are correctly classified:

$$\texttt{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

Recall (Rec.) measures the proportion of actual positive samples that are correctly identified and is crucial for minimizing missed diagnoses:

$$\texttt{Recall} = \frac{TP}{TP + FN} \tag{2.10}$$

Precision (Prec.) measures the proportion of positive predictions that are correct, indicating the reliability of a positive diagnosis:

$$\texttt{Precision} = \frac{TP}{TP + FP} \tag{2.11}$$

Specificity (Spec.) measures the proportion of actual negative samples that are correctly identified, reflecting the model's ability to rule out the condition:

$$\texttt{Specificity} = \frac{TN}{TN + FP} \tag{2.12}$$

F1-Score (F1) is the harmonic mean of precision and recall, providing a balanced measure of a model's performance, which is especially useful in cases of class imbalance:

$$\texttt{F1-Score} = 2 \cdot \frac{\texttt{Precision} \cdot \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} = \frac{2TP}{2TP + FP + FN} \tag{2.13}$$

AUPRC. The Area Under the Precision–Recall Curve (AUPRC) is a threshold-independent metric. It summarizes the trade-off between precision and recall across all possible thresholds. The AUPRC is particularly informative for imbalanced datasets, as it focuses on the performance of the minority (positive) class and is less influenced by the large number of true negatives than the AUROC.

Statistical comparison. To formally compare the models using each metric, we estimated 95% confidence intervals for the between-model difference ($\Delta = \text{metric}_{\text{Slider}} - \text{metric}_{\text{baseline}}$) using a class-stratified, paired bootstrap at the patient level ($B = 5000$ resamples; the same resampled indices were applied to both models). Two-sided p-values were obtained via a within-case score-swapping permutation test for the AUPRC and DeLong's test for the AUROC. All tests were two-sided with $\alpha = 0.05$.

Comparison methods. For the PM extractor, we evaluate several model families, including ResNet [84], DenseNet [85], and EfficientNet [82]. For the Slider, we compare our method against several transfer learning baseline methods:

- **Full fine-tuning:** fully updates all parameters of the backbone for PF-ILD identification.
- **Partial fine-tuning:** updates only the last ViT layer while keeping all other layers frozen.

- **Linear probing:** trains only the linear classification layer, keeping all other parameters fixed.

2.4.2 Results of PSM-Based PM Extractor

We evaluate the performance of various backbone models for the PSM-based PM extractor on the same training and testing datasets using the 1-Up-Down Accuracy metric. The models span several architecture families, including ResNet [84], DenseNet [85], and EfficientNet [82], and the best-performing model from each family is summarized in Table 2.2.

Table 2.2: The results of the PSM-based PM extractor with 1-Up-Down accuracy. The best scores for each PM are highlighted in **bold**.

Model	Upper PM (%)	Middle PM (%)	Lower PM (%)	Avg. (%)
ResNet-101	98.33	96.67	95.0	96.67
DenseNet-169	**100**	**98.33**	95.0	97.78
EfficientNet-b4	**100**	**98.33**	**96.67**	**98.33**

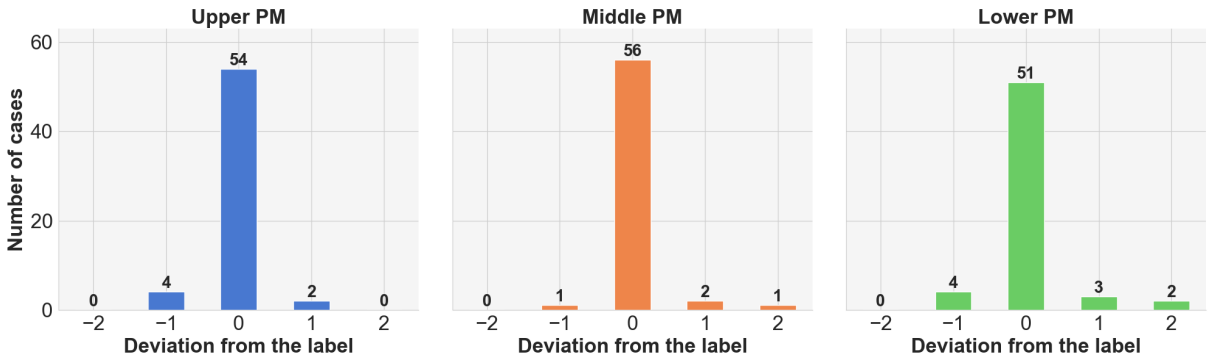


Figure 2.4: Statistics on the extracted slices from the PSM-based PM extractor on EfficientNet-b4.

Among all the models, EfficientNet-b4 achieves the highest overall performance, with an average 1-Up-Down Accuracy of 98.33%. Its class-wise performance is also strong, reach-

ing 100% for the upper PM, 98.33% for the middle PM, and 96.67% for the lower PM extraction. Within the DenseNet family, DenseNet-169 performs the best, achieving an average 1-Up-Down Accuracy of 97.78%. Its accuracy for the upper and middle PM classes is comparable to that of EfficientNet-b4. These results suggest that the upper PM is a relatively easy prediction target, likely because it corresponds to the slice just before the lung fields become visible, a visually distinct and consistent anatomical feature. In contrast, identifying the lower PM is more challenging as there is greater variability in lung morphology near the diaphragm across patients.

To further validate the effectiveness of the PSM-based PM extractor, we analyze the deviation of predicted slices from ground-truth PMs using EfficientNet-b4, as shown in Figure 2.4. We find that the majority of predictions are either exactly correct or within one slice of the ground truth. Notably, even incorrect predictions remain within two slices above or below the reference PM, demonstrating the robustness and reliability of the proposed PSM-based approach.

2.4.3 Results on Slider for PF-ILD Identification

Table 2.3 summarizes the PF-ILD identification performance of the proposed Slider model under three different RS configurations. In the 5-RS setting, Slider achieves the following results: $\mathrm{AUPRC} = 0.790$ (95% CI [0.652, 0.901]) and $\mathrm{AUROC} = 0.847$ (95% CI [0.760, 0.921]). This corresponds to improvements of $+0.008$ and $+0.033$ over full fine-tuning ($\mathrm{AUPRC} = 0.782$, 95% CI [0.645, 0.894]; $\mathrm{AUROC} = 0.814$, 95% CI [0.706, 0.907]). As detailed in the Methods (statistical comparison) section, we compared the AUROC using DeLong's test, yielding the following result: $\Delta\mathrm{AUROC} = 0.033$ (95% CI $[-0.049, 0.115]$; $z = 0.783$; $p = 0.434$). We compared the AUPRC using a class-stratified, paired bootstrap of the difference (Δ) to obtain 95% CIs, with $\Delta\mathrm{AUPRC} = 0.008$ (95% CI $[-0.073, 0.086]$), and computed a within-case score-swapping permutation p-value ($p = 0.854$). All tests were two-sided at the $\alpha = 0.05$ significance level. Overall, the point estimates favor Slider, but the differences are not statisti-

Table 2.3: Performance of Slider on the PF-ILD datasets across different metrics for 3-RS, 5-RS, and 9-RS settings. #Params denotes the number of tunable parameters. Rows shaded in light blue indicate our method. The best score for each setting is shown in **bold**.

Method	RSs	#Params (M)	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
Full Fine-tuning	×3	86.58	0.720	0.757	0.622	0.696	0.683	0.692	0.797
Partial Fine-tuning	×3	7.09	0.742	0.811	0.638	0.696	0.714	**0.738**	0.810
Linear Probing	×3	0.0015	0.602	**0.946**	0.500	0.375	0.654	0.726	0.780
Slider	×3	3.56	**0.785**	0.703	**0.743**	**0.839**	**0.722**	0.735	**0.813**
Full Fine-tuning	×5	86.58	0.656	**0.838**	0.544	0.536	0.660	0.782	0.814
Partial Fine-tuning	×5	7.09	0.688	**0.838**	0.574	0.589	0.681	0.760	0.832
Linear Probing	×5	0.0015	0.785	0.730	0.730	0.821	0.730	0.724	0.774
Slider	×5	3.56	**0.796**	0.730	**0.750**	**0.840**	**0.740**	**0.790**	**0.847**
Full Fine-tuning	×9	86.58	**0.763**	0.541	**0.800**	0.911	0.645	**0.771**	0.808
Partial Fine-tuning	×9	7.09	**0.763**	**0.811**	0.667	0.732	**0.732**	0.712	**0.823**
Linear Probing	×9	0.0015	0.731	0.432	**0.800**	**0.929**	0.561	0.718	0.768
Slider	×9	3.56	0.731	0.703	0.650	0.750	0.675	0.751	0.821

cally significant; notably, Slider demonstrates comparable performance with substantially fewer trainable parameters (3.56 M), signifying its computational efficiency and deployability.

Across all configurations, 5-RS yields the best overall performance, followed by 9-RS and then 3-RS. Notably, in the 5-RS setting, Slider outperforms full fine-tuning on nearly all metrics except recall, highlighting its strong parameter efficiency and effectiveness. Partial fine-tuning achieves the second-highest AUROC (0.832), confirming that lightweight adaptation remains competitive. Linear probing achieves relatively high recall (0.730) but suffers from low precision and specificity, resulting in a lower AUPRC (0.724) and AUROC (0.774). These results demonstrate that Slider achieves the best trade-off between model complexity and diagnostic accuracy, making it particularly well suited for realistic clinical deployment.

2.4.4 Domain Shift Analysis

In practical applications, domain shifts frequently occur, representing variations between the training dataset, $\mathcal{S}$, and the target environment. These discrepancies reduce performance when models are applied beyond their original training distribution. Such shifts are prevalent across datasets and are widely used as benchmarks to evaluate the robustness of machine learning models [86]. To investigate ILD-Slider's resilience to domain shifts, we train Slider models on datasets from different facilities and evaluate them using the AUROC with $\delta = 4$. The results are summarized in Table 2.4.

In general, datasets from different facilities exhibit distinct data regimes. The OTMC is less heterogeneous, so Slider can learn it more easily, while the UOH dataset is more heterogeneous. Models trained on a single facility show strong in-domain performance but suffer from notable degradation when tested on data from a different facility. For example, the Slider trained on the UOH dataset achieves an AUROC of 0.750 on this data, but the AUROC increases to 0.896 on OTMC data, whereas the model trained on the OTMC dataset attains 0.921 on this data, yet the value drops to 0.759 on UOH data. In contrast, the model trained on both the UOH and OTMC datasets demonstrates the most balanced and robust behavior, achieving AUROCs of 0.823 on UOH data, 0.875 on OTMC data, and 0.847 on the combined UOH and OTMC test set. These findings underscore the importance of multi-facility training for Slider, which is critical for reliable PF-ILD identification across different scanners and institutions.

Table 2.4: Cross-domain evaluation of Slider on AUROC. Each row corresponds to the facility used for training, and each column corresponds to the facility used for testing.

Domain	**UOH**	**OTMC**	**UOH and OTMC**
UOH	0.750	0.896	0.791
OTMC	0.759	0.921	0.803
UOH and OTMC	0.823	0.875	0.847

2.4.5 Visualization of TPs, TNs, FPs, and FNs

To better understand the behavior of the proposed model in PF-ILD identification, representative TP, TN, FP, and FN examples are shown in Figure 2.5, using Grad-CAM [87] to highlight attention maps for the predicted class. For TP cases, Slider effectively captures fibrosis-related features across slices, enabling accurate PF-ILD identification. For TN cases, the model predominantly focuses on non-lung regions, consistent with the absence of representative fibrosis in lung tissue. In FP cases, however, Slider incorrectly highlights fibrotic-like patterns within the lung field; although fibrosis is present, it is insufficient to confirm PF-ILD, leading to misclassification. These instances illustrate the clinical difficulty of distinguishing incidental fibrosis from PF-ILD. FN cases further reveal model limitations. For OTMC data, Slider mistakenly attends to non-lung regions (e.g., surrounding tissues) while overlooking fibrosis within the lungs. Similarly, for UOH data, the model becomes confused when subtle fibrosis appears in the lung field, leading to missed PF-ILD identification. These examples emphasize the challenges of PF-ILD detection in real-world clinical settings.

2.4.6 Sensitivity to Dataset Size

To investigate whether Slider is sensitive to the size of the training set, $\mathcal{S}$, we train it on six subsets with different ratios (0.1, 0.2, 0.4, 0.6, 0.8, and 1.0). The results are shown in Figure 2.6. Overall, Slider consistently outperforms full fine-tuning across all settings, demonstrating its robustness to varying dataset sizes. Both methods benefit from larger training sets, but Slider exhibits a particularly notable improvement when the ratio increases from 0.2 to 0.4. In contrast, full fine-tuning shows optimal performance gain when this ratio changes from 0.4 to 0.6. Importantly, Slider requires only 60% of the training data to surpass full fine-tuning across all dataset sizes, highlighting both its parameter efficiency and its superior data efficiency.

2.4.7 Further Analyses on ILD-Slider

This section further investigates the capabilities of ILD-Slider through a series of experiments.

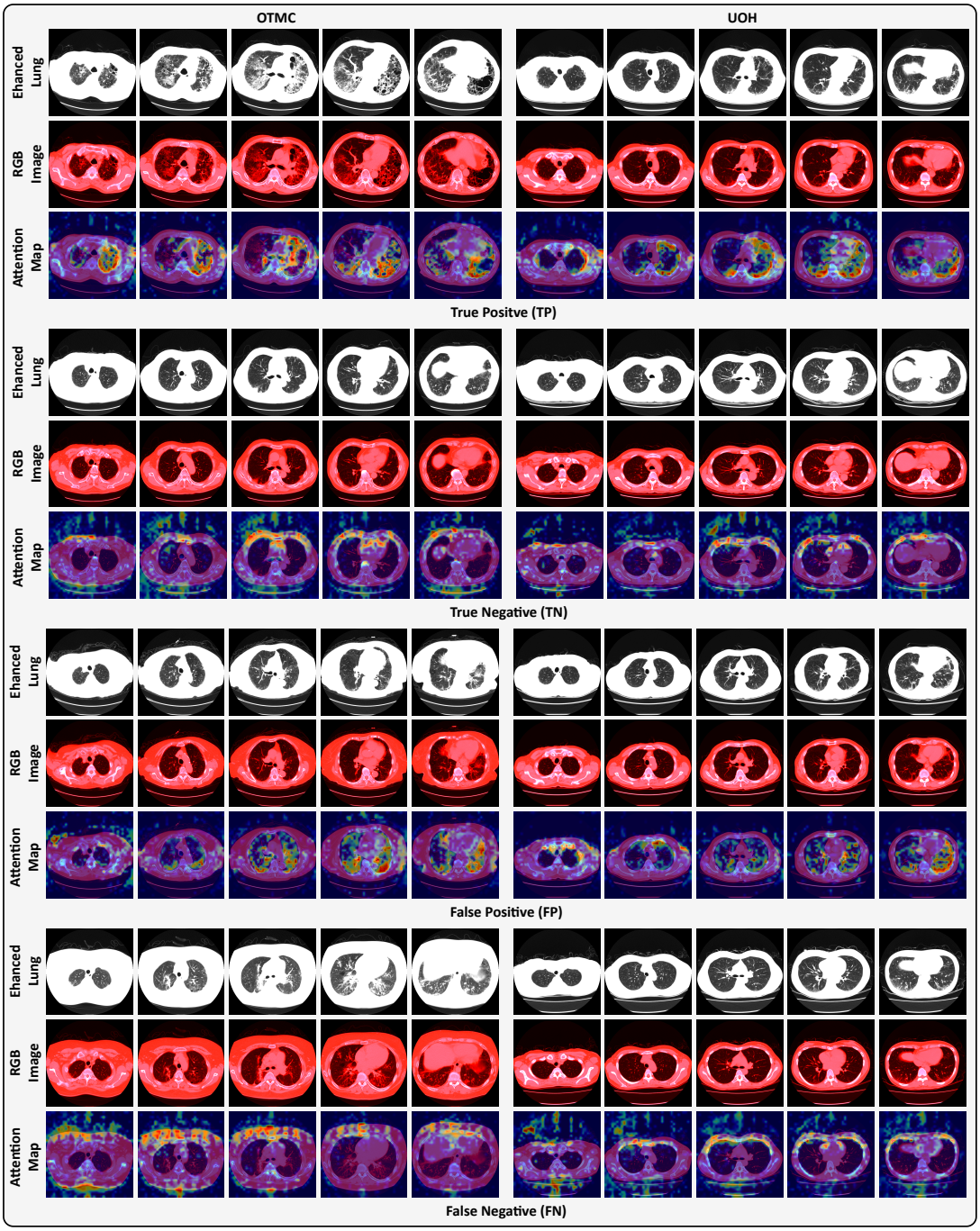


Figure 2.5: Visual examples under the 5-RS setting of TP, TN, FP, and FN cases from the OTMC and UOH datasets. Each case presents the enhanced lung window image (Enhanced Lung), the RGB windowing-processed image (RGB Image), and the Grad-CAM attention map (Attention Map) corresponding to the predicted class.

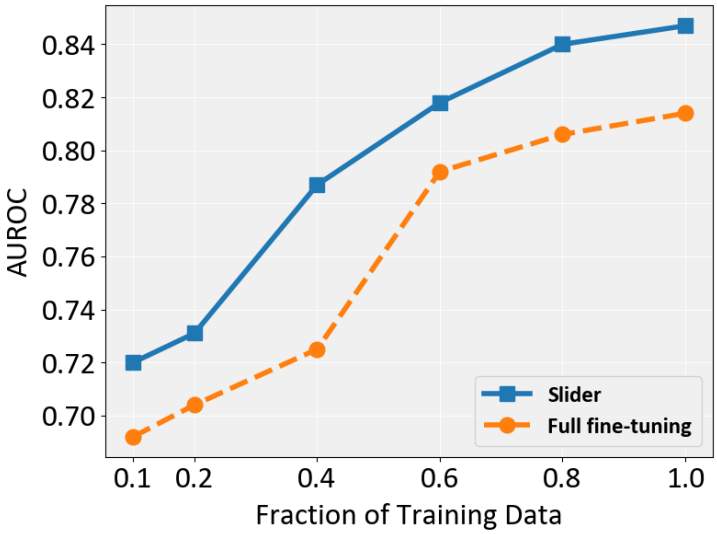


Figure 2.6: Performance comparison of full fine-tuning and Slider in terms of AUROC across different training data fractions.

The dimension on which the Slider should be applied. We set the adapter dimension, p, to 192 in Equation (2.5), which corresponds to a scale factor of $\delta = 4$. To systematically investigate the impact of different scale factors δ, we evaluate $\delta \in \{1, 2, 4, 6, 8, 12\}$ (corresponding to $p = 768, 384, 192, 128, 96, 64$, respectively). The results are summarized in Table 2.5.

We found that the best performance is achieved at $\delta = 4$, with an AUPRC of 0.790 and an AUROC of 0.847. Moreover, this setting provides the most favorable balance between recall and precision. In contrast, both smaller ($\delta = 2$) and larger ($\delta = 12$) scale factors yield decreased performance. Notably, a parameter efficiency of $\delta = 4$ (3.56 M tunable parameters) outperforms both higher-capacity ($\delta = 1$) and lower-capacity settings ($\delta = 12$), highlighting an effective trade-off between model capacity and diagnostic accuracy in Slider.

The impact of RGB windowing processing. RGB windowing applies different window levels and widths to emphasize tissue-specific features, enabling the Slider model to capture a richer set of visual cues. Table 2.6 compares Slider's performance with and without RGB windowing. Without RGB windowing, its performance markedly drops across most metrics: the AUROC decreases from 0.847 to 0.808, the AUPRC from 0.790 to 0.762, and specificity from

Table 2.5: The performance of Slider for different scale factors, δ. The row shaded in light blue indicates our default settings.

Setting	#Params (M)	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
$\delta = 1$	14.21	0.710	0.622	0.639	0.768	0.630	0.630	0.811
$\delta = 2$	7.11	0.720	0.730	0.628	0.714	0.675	0.723	0.806
$\delta = 4$	3.56	0.796	0.730	0.750	0.840	0.740	0.790	0.847
$\delta = 6$	2.38	0.710	0.730	0.614	0.696	0.667	0.742	0.797
$\delta = 8$	1.79	0.774	0.730	0.711	0.804	0.720	0.777	0.816
$\delta = 12$	1.19	0.785	0.649	0.774	0.875	0.706	0.770	0.837

0.840 to 0.446, indicating a sharp increase in false positives. Although recall increases from 0.730 to 0.919 due to the model generating more false positive predictions, this comes at the cost of reduced precision (0.750 to 0.523) and overall accuracy (0.796 to 0.634).

Table 2.6: The performance of Slider with (w/) and without (w/o) RGB windowing processing. The row shaded in light blue indicates our default settings.

Setting	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
w/o RGB	0.634	0.919	0.523	0.446	0.667	0.762	0.808
w/ RGB	0.796	0.730	0.750	0.840	0.740	0.790	0.847

These results demonstrate that RGB windowing yields more balanced and robust diagnostic performance by enhancing the visibility of fibrotic regions while preserving discriminative power for both positive and negative classes in PF-ILD identification.

The impact of using representative slices. We support the use of RSs for PF-ILD identification because they capture anatomically consistent lung parenchyma regions determined by PMs. To evaluate their effectiveness, we compare the proposed Slider model using RSs with

randomly selected slices across three runs with different random seeds, as shown in Table 2.7.

Table 2.7: The performance of Slider on randomly selected slices (mean $\pm$ std over three runs with different random seeds) and RSs. The row shaded in light blue indicates our method.

Method	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
Random	0.738 ± 0.05	0.730 ± 0.09	0.659 ± 0.08	0.744 ± 0.08	0.689 ± 0.06	0.739 ± 0.03	0.817 ± 0.02
Slider	0.796	0.730	0.750	0.840	0.740	0.790	0.847

The results indicate a clear and consistent advantage when using RSs. Compared to random slices, RSs improve the AUROC from 0.817 to 0.847 and the AUPRC from 0.739 to 0.790, reflecting stronger overall discrimination and more reliable positive class predictions. Similarly, the F1-Score increases from 0.689 to 0.740. The specificity also increases from 0.744 to 0.840, indicating that RSs help reduce false positives. These improvements demonstrate that PM-guided RS selection not only enhances sensitivity to disease-relevant regions but also minimizes noise from non-informative slices, leading to more accurate and robust PF-ILD identification with Slider.

Effect of kernel shape in Slider. To assess the impact of incorporating slice-level information in Slider for PF-ILD identification, we evaluated different 3D convolution kernel shapes (Table 2.8). The results show that slice-level modeling plays a crucial role in achieving strong diagnostic performance. When the kernel only captures spatial context without inter-slice information ($1 \times 3 \times 3$), the AUROC drops to 0.797 and the AUPRC to 0.718, indicating reduced discriminatory ability. Conversely, purely slice-wise kernels (our default setting) without spatial aggregation ($3 \times 1 \times 1$) achieve the best results, with an AUROC of 0.847 and an AUPRC of 0.790, suggesting that inter-slice context is more critical than additional spatial filtering for PF-ILD identification.

The $3 \times 3 \times 3$ kernel, which combines both spatial and slice-level information, yields a competitive AUROC (0.843) but underperforms compared to the $3 \times 1 \times 1$ kernal, possibly due to fine-grained slice-level patterns that are relevant to disease progression being over-smoothed.

Table 2.8: The performance of Slider on different kernel shape settings. The row shaded in light blue indicates our default settings.

Kerneal Shape	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
$1 \times 1 \times 1$	0.591	0.919	0.493	0.375	0.642	0.784	0.809
$1 \times 3 \times 3$	0.753	0.568	0.750	0.875	0.646	0.718	0.797
$3 \times 3 \times 3$	0.763	0.811	0.667	0.732	0.732	0.749	0.843
$3 \times 1 \times 1$	0.796	0.730	0.750	0.840	0.740	0.790	0.847

The $1 \times 1 \times 1$ kernel, lacking both spatial and slice context, performs worst, confirming that contextual cues, particularly along the slice dimension, are indispensable for PF-ILD identification with Slider.

Table 2.9: The performance of single- and multi-layer linear probing on 5-RS.

Setting	#Params (M)	Acc.	Rec.	Prec.	Spec.	F1	AUPRC	AUROC
Single-layer	0.0015	0.785	0.730	0.730	0.821	0.730	0.724	0.774
Two-layer	0.3948	0.753	0.595	0.733	0.857	0.657	0.774	0.802
Three-layer	0.5256	0.720	0.649	0.649	0.768	0.649	0.740	0.786

Comparison of multi-layer linear probing. As shown in Table 2.3, single-layer linear probing achieves competitive performance compared with other baselines, while requiring only 0.0015M tunable parameters. To examine whether deeper probe heads further improve performance, we extend the probe to two-layer and three-layer Multilayer Perceptrons (MLPs). The results are shown in Table 2.9. The two-layer setting improves AUPRC and AUROC over the single-layer probe, whereas the three-layer setting leads to a performance drop compared with the two-layer probe. The multi-layer setting introduces more trainable parameters than the single-layer probe, though it remains significantly fewer than Slider. Nevertheless, even with this increased capacity, it still falls short of Slider's overall performance.

2.5 Discussion

Clinical translation. ILD-Slider can efficiently identify suspected PF-ILD cases from a limited number of RSs, supporting its prioritization for longitudinal evaluation and further examinations. It has the potential to facilitate screening and reading, particularly in settings with limited access to expert radiologists. However, the final clinical decision should always rely on both integration of longitudinal assessments and clinical findings.

Longitudinal label interpretation. Although the outcome is defined longitudinally, a single-time point HRCT can encode prognostic information. Prior studies [88–90] have shown that baseline fibrotic burden and HRCT features are predictive of progression and mortality in fibrosing ILD. We therefore hypothesize that ILD-Slider primarily relies on a fibrosis-weighted parenchymal texture signature, favoring reticulation and traction bronchiectasis over transient or inflammatory ground-glass changes, as well as on apex-to-base (craniocaudal) gradients captured by RSs. These imaging characteristics are commonly observed in fibrosing ILD, particularly when a UIP-like pattern is present (e.g., basal- and subpleural-predominant fibrosis with an apicobasal gradient).

Limitations. In this study, PF-ILD labels are assigned based on clinical diagnostic criteria derived from retrospective medical records, including $\geq$10% fibrosis, a decline in FVC, and evidence of clinical and radiological progression. Accordingly, estimations made solely from cross-sectional imaging findings are, by definition, subject to an inevitable degree of discrepancy. In particular, longitudinal declines in respiratory function may encompass aspects that are not directly observable through imaging, thereby representing a potential factor that constrains the upper bound of the model's performance. Furthermore, our dataset was collected from two hospitals within the same geographic region in Japan. It may limit the generalizability of ILD-Slider to other populations, scanners, or clinical settings.

Future work. To overcome the inherent limitations of imaging alone, future work should explore multimodal modeling that incorporates pulmonary function metrics, such as FVC, clinical symptoms, and biomarkers, or directly train models on disease progression using longitu-

dinal data. In addition, important next steps in this field include external calibration and validation, decision curve analysis to evaluate clinical utility, and the development of radiologist-facing user interfaces with enhanced explainability.

2.6 Summary

We propose a new method, ILD-Slider, which is a parameter-efficient and lightweight framework for PF-ILD identification from a limited number of CT slices. By introducing anatomy-based PMs and a PSM strategy, ILD-Slider effectively selects RSs, substantially reducing computational cost while preserving diagnostic accuracy. Extensive experiments on datasets from two independent medical facilities demonstrated that ILD-Slider achieves robust performance (an AUPRC of 0.790 and an AUROC of 0.847 with only five RSs), confirming its practicality for real-world clinical settings. Furthermore, our analysis highlights the benefits of RGB windowing processing, cross-domain generalization, and the feasibility of diagnosing PF-ILD from non-contiguous slices, representing an important step toward broader applicability on public datasets. These findings underscore ILD-Slider as both a data-efficient and computationally efficient solution for early PF-ILD identification.

Chapter 3

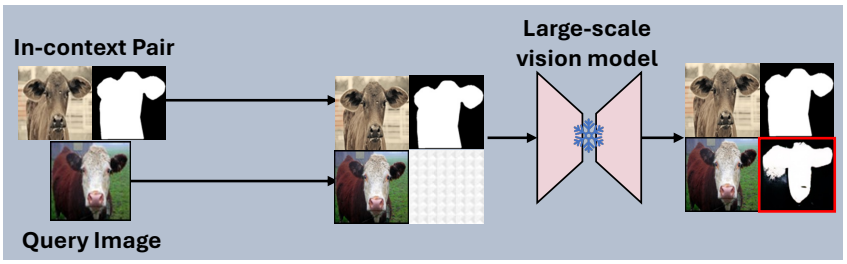
E-InMeMo: Pixel-Level In-Context Pair Enhancement for VICL

3.1 Overview

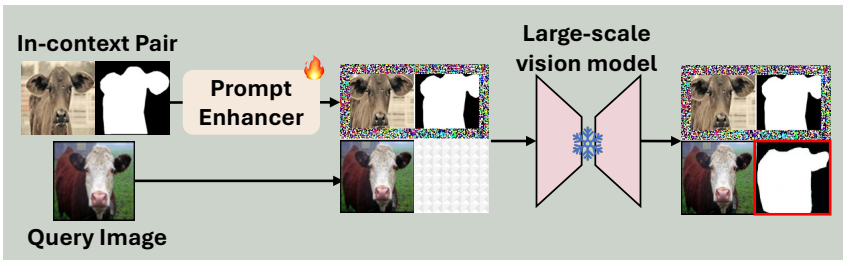
Recent years have witnessed significant progress in large-scale models, which have shown impressive generalization capabilities and strong potential across various downstream tasks [8, 14, 77]. Representative models like GPT [6] and Gemini [7] exemplify the effectiveness of *in-context learning* (ICL) in Natural Language Processing (NLP) [32–39, 52]. Through ICL, models can tackle novel tasks by leveraging prompt-based inputs to infer predictions on previously unseen data, without requiring updates to their parameters, thus substantially reducing training costs. Despite its promise as a foundational paradigm for deploying large-scale models in real-world scenarios, the exploration of ICL in computer vision applications is still at a relatively early stage [10, 31, 40].

MAE-VQGAN [10] marks a groundbreaking development by showcasing the potential of applying ICL to a range of computer vision tasks, such as image segmentation, inpainting, and style transfer. This method introduces visual prompts arranged in a four-cell grid canvas, as shown in Figure 3.1a, which includes a query image alongside an input–output example known

as an in-context pair. These in-context pairs illustrate the task at hand by pairing an input image with its corresponding label or output image. Prior work [31] has emphasized the importance of such pairs in effectively guiding the model toward producing accurate results. In visual ICL, it is particularly important that the in-context image closely resembles the query image in terms of semantics, perspective, and other attributes [31], as illustrated in Figure 3.2. Consequently, selecting appropriate in-context pairs becomes a critical component of the overall process.



(a) Visual in-context learning



(b) E-InMeMo

Figure 3.1: A schematic comparison between standard visual ICL and E-InMeMo. (**a**) Visual ICL constructs a four-cell grid canvas composed of a query image, an in-context pair, and an empty cell (bottom-right) where the model generates the prediction. This canvas serves as the *prompt*, and the output (marked in the red box) is produced by passing it through a frozen large-scale vision model. (**b**) E-InMeMo enhances this paradigm by introducing a *learnable prompt*, a trainable perturbation designed to adjust the distribution of the in-context pair, thereby improving task guidance and prediction accuracy.

Although visual ICL methods have shown promising results [31, 91], the in-context pairs retrieved are not always optimal. This is often due to limitations in the available retrieval dataset and a mismatch between the prompt content and the latent knowledge embedded within large-

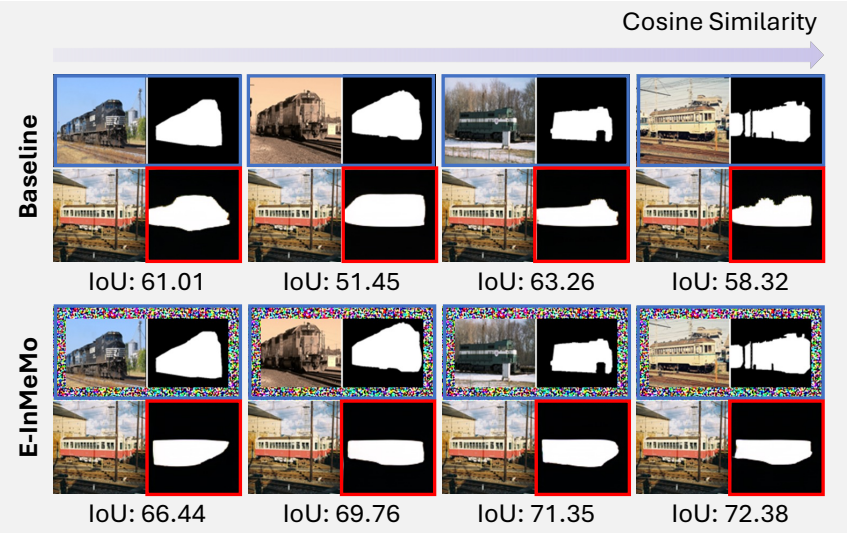


Figure 3.2: Performance comparison of visual ICL on a foreground segmentation task. Blue boxes indicate in-context pairs, while red boxes represent the predicted label images (query images are unmarked). The quality and similarity of the in-context pair significantly influence the prediction results. In the absence of a learnable prompt, performance is highly dependent on the semantic closeness between the query and in-context images. In contrast, E-InMeMo, with its learnable prompt, produces more stable and accurate predictions across varying input conditions.

scale vision models. This raises a critical question: Is it possible to refine or transform the prompt to better guide the model for downstream visual ICL tasks?

The concept of pixel-level learnable prompting [23,67,92] provides a promising solution. It adjusts the model's input without altering the model parameters, enabling adaptation to various tasks. Unlike the adapter-based methods discussed in Chapter 2 which modify the model architecture, this method focuses on input-level optimization. It is especially suitable for large models where conventional fine-tuning can be prohibitively expensive due to their scale [67,92–94]. Learnable prompts have demonstrated a remarkable ability to adapt, even in cases with significant semantic or distributional differences [23,67,92].

As the second work of this thesis exploring input-level optimization, we introduce a novel

visual ICL method called **E**nhanced **In**struct **Me** **Mo**re (**E-InMeMo**), which leverages learnable visual prompts to instruct large vision models. After retrieving an initial in-context pair, we refine it using a dedicated prompt enhancer inspired by [67]. As with previous visual ICL approaches, the refined in-context pair is combined with the query into a unified image canvas, which is then processed by a pre-trained large-scale vision model [10]. The learnable prompt is trained in a supervised fashion to produce the correct label image for the query.

Contributions. E-InMeMo offers a data- and parameter-efficient solution that enables efficient training while achieving high performance. By customizing a learnable prompt for a given task, the framework reshapes the prompt distribution to be more aligned with the task objectives and bridges the visual discrepancies between query and in-context examples. Our experiments establish new state-of-the-art (SOTA) results in tasks such as foreground segmentation and single object detection. Although E-InMeMo requires training, it substantially mitigates the limitations posed by suboptimal visual prompts.

This chapter extends our earlier conference work published at WACV 2024 [43]. The enhancements in this version include the following:

- We introduce a new strategy to apply learnable prompt directly to the in-context pair, with strong robustness.
- We remove the need for resizing the in-context pairs, and thus reducing trainable parameters from 69,840 to 27,540.
- We replace CLIP [8] with DINOv2 [9] for feature extraction, improving representation quality.
- We validate E-InMeMo's consistent outperformance across diverse datasets, including natural and medical images. Furthermore, we present comprehensive experiments to highlight E-InMeMo's robustness and generalizability.

3.2 Related Work

3.2.1 In-Context Learning

In-Context Learning (ICL) has emerged as a powerful framework in Natural Language Processing (NLP), particularly with the advent of large language models (LLMs) such as GPT-3 [30]. By presenting several input–output examples pertaining to a specific task, this paradigm enables autoregressive models to perform inference without adjusting their parameters [31]. ICL has gained significant traction due to its various strengths [95], including an intuitive and interpretable user interface [30, 96, 97], its resemblance to human-like reasoning [98], and its suitability for plug-and-play deployment as a service [99]. It has opened doors to new possibilities across different areas [100–102], such as solving complex reasoning tasks [103], question answering [101, 104], and compositional generalization [105, 106].

In the computer vision field, ICL has sparked growing interest [10, 31, 40, 42, 75, 76]. Unlike NLP, where instructions are text-based, visual ICL presents a unique challenge: how to clearly define the task for the model using visual inputs. Bar et al. [10] introduced a novel method that combines an input–output image pair with a query image, to specify the task. This setup, presented as a single image, reframes the task as a form of image inpainting task. Subsequent efforts like Painter [40], which adopts the MAE architecture instead of VQGAN, and SegGPT [41], which generalizes the method to image and video segmentation, have built upon this setting. LVM [42] further pushes the boundaries by tokenizing visual inputs and applying an autoregressive learning strategy akin to that used in LLMs.

Beyond developing new architectures, recent studies underscore the critical role of in-context image quality in determining performance [31, 91]. Zhang et al. [31] proposed a supervised approach for prompt selection, emphasizing that both the number and the relevance of in-context pairs significantly affect results. Similarly, Sun et al. [91] explored pixel-level retrieval methods for selecting in-context pairs, experimenting with multiple layout configurations to improve ICL effectiveness.

While existing research has firmly established the importance of in-context pairs for down-

stream tasks [10, 31, 91], little attention has been paid to altering or enhancing these pairs to boost visual ICL performance. Our study seeks to fill this gap by introducing learnable perturbations to refine in-context pairs and ultimately enhance downstream outcomes.

3.2.2 Learnable Prompting

In the NLP field, prompting is widely used to adapt LLMs for specific downstream applications [16]. For example, while GPT-3 [30] demonstrates strong generalization across many tasks, it often relies on carefully engineered prompts, an approach that can be labor-intensive. Additionally, fully fine-tuning these massive models incurs high computational costs. Parameter-efficient fine-tuning (PEFT) addresses these challenges by modifying only a small fraction of the model parameters or by adding auxiliary components, such as adapters [15, 17, 18] or learned prompt vectors [19, 20], achieving performance comparable to full fine-tuning with far less overhead.

Motivated by the success of PEFT in NLP, researchers have extended these strategies to computer vision [21–24] and vision-language models [8, 25–27]. A common approach involves augmenting the input image with learnable prompt embeddings or perturbations. Several recent methods have demonstrated how trainable modifications to visual inputs can enhance model performance [67, 92, 107]. For instance, Bahng et al. [67] proposed adding learnable, input-agnostic pixel-level prompts to input images to better adapt large, frozen models like CLIP [8] to new tasks. ILM-VP [92] extended this by introducing task-aware prompts that incorporate label mapping for improved adaptation. Since this method involves optimizing only a small number of parameters relative to the full model, it offers a practical and scalable solution, particularly well-suited for integration into visual ICL pipelines.

In this work, we extend the concept of learnable pixel-level prompting to the VICL paradigm. While previous methods [67, 92] demonstrated the efficacy of visual prompting for adapting models to specific tasks, we explore this strategy to the VICL paradigm to mitigate the visual discrepancies between queries and in-context pairs. Specifically, we demonstrate that optimizing learnable perturbations for in-context pairs can effectively bridge the visual discrepancies

between the query and the in-context pair. To the best of our knowledge, this study provides the first empirical evidence that pixel-level visual prompting serves as a data- and parameter-efficient strategy to enhance the inference capabilities of frozen foundation models in the VICL setting.

3.3 Method

3.3.1 Preliminary: MAE-VQGAN

MAE-VQGAN represents a pioneering effort that first introduces visual ICL via inpainting [10]. Consider a query image denoted by $x_{\mathrm{q}} \in \mathbb{R}^{C\times H\times W}$, alongside an example input $x \in \mathbb{R}^{C\times H\times W}$ and its corresponding ground-truth output $y \in \mathbb{R}^{C\times H\times W}$ (e.g., a segmentation result). These components form what is referred to as *prompt*. The prompt is created by concatenating x_{q}, x, and y into a single canvas $c_{\mathrm{q}} = [x, y, x_{\mathrm{q}}, r]$. Following [10], we introduce a two-pixel gap between images, resulting in a canvas dimensions of $\mathbb{R}^{C\times 2(H+1)\times 2(W+1)}$, where $[\cdot,\cdot,\cdot,\cdot]$ represents image concatenation to formulate a four-cell grid canvas. The region $r \in \mathbb{R}^{C\times H\times W}$ remains blank, designated for the task output $\hat{y}_{\mathrm{q}}$[1]. The region r is subdivided into L patch regions, forming a patch set $\mathcal{L} = \{r_l\}_{l=1}^{L}$. The MAE-VQGAN computes the assignment scores vector for the patch region r_l as follows:

$$\mathbf{s}_l = g_l(c_{\mathrm{q}}) \quad \in \mathbb{R}^{|\mathcal{V}|}, \tag{3.1}$$

where $\mathcal{V}$ represents the pre-trained VQGAN codebook, and g_l is the MAE function that outputs the assignment score of r_l, with each element $s_{lv} \in \mathbf{s}_l$ indicating the score for visual token $v \in \mathcal{V}$. These scores can be interpreted as the probability $p(w_l = v|c_{\mathrm{q}})$ that the visual token assigned to l, denoted by w_l, is v. After determining the appropriate visual token for l as

[1] In [10], the arrangement of x_{q}, x, and y is flexible. A mask is used to specify the region r to be filled. For brevity, these details are omitted here.

$z_l^\star = \arg\max_v s_{lv}$, the VQGAN decoder f generates the prediction $\hat{y}_q$, expressed as follows:

$$\hat{y}_q = f(\{z_l^\star\}_{l=1}^L) \quad \in \mathbb{R}^{C\times H\times W}. \tag{3.2}$$

3.3.2 Overview of E-InMeMo

Let $\mathcal{S} = (x, y)$ represent a dataset consisting of input images x paired with their corresponding label (or output) images y, designated for a specific downstream task, where $|\mathcal{S}| = n$. Given this dataset and a query image x_q, the goal is to predict the output y_q for the query.

An outline of the proposed E-InMeMo framework is illustrated in Figure 3.3. First, the query image x_q is passed through an retriever, which selects a similar in-context pair (x, y) from the dataset $\mathcal{S}$. This selected pair is combined with the query image into a 2×2 grid format, forming what we refer to as a canvas. The canvas is expressed as $c_\mathrm{q} = [x, y, x_\mathrm{q}, r]$, where r denotes an empty cell.

The prompt enhancer, which integrates learnable prompt, then refines the in-context pair to produce (x', y'). This results in an enhanced version of the canvas: $\hat{c}_\mathrm{q} = [x', y', x_\mathrm{q}, r]$. This enhanced canvas is then input to a frozen, pre-trained large-scale vision model g, which outputs a sequence of visual tokens $\hat{\mathbf{z}} = g(\hat{c}_\mathrm{q})$. Simultaneously, the complete ground-truth canvas $c_\mathrm{gt} = [x, y, x_\mathrm{q}, y_\mathrm{q}]$ is encoded using a VQGAN encoder to generate the reference visual tokens $\mathbf{z} = g(c_\mathrm{gt})$.

To optimize the learnable prompt, we compute a cross-entropy loss over the visual tokens corresponding to the empty cell location. These tokens encode the model's prediction $\hat{y}_\mathrm{q}$, which is decoded by a function f that maps tokens back into the pixel domain, producing the final output $\hat{y}_\mathrm{q} = f(\hat{\mathbf{z}})$.

At the heart of E-InMeMo lies the prompt enhancer, denoted as t_ϕ, parameterized by a learnable set ϕ. This component is trained on dataset $\mathcal{S}$ to generate instructive and task-specific transformations, ensuring effective guidance even in scenarios where the retrieved in-context pair is suboptimal.

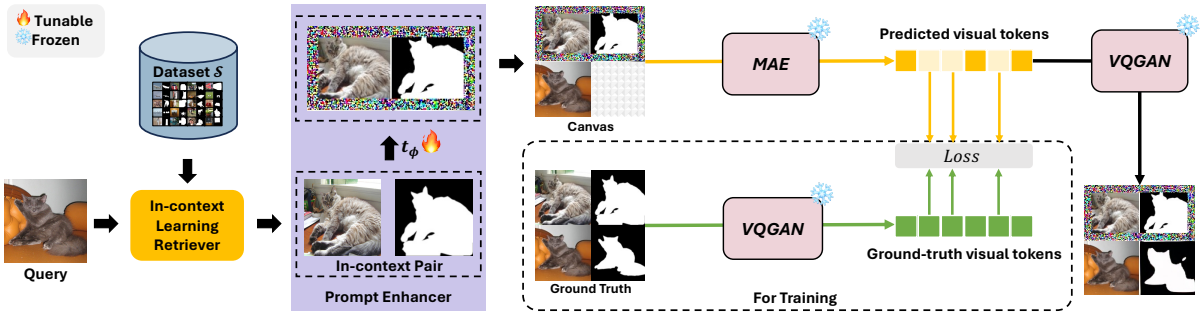


Figure 3.3: Overview of the proposed E-InMeMo framework. The process begins with the In-context Learning Retriever, which selects an in-context pair from the dataset $\mathcal{S}$ for a given query image. A Prompt Enhancer $t_\phi(\cdot)$ then applies learnable perturbations to the in-context pair, producing an enhanced version. These enhanced in-context images, along with the query and an empty cell (bottom-right), are arranged into a four-cell grid canvas. This canvas is passed through a frozen `MAE`, which outputs predicted visual tokens corresponding to the empty cell. For visualization, the predicted tokens are decoded into an image using the decoder of `VQGAN`. During training, a ground-truth canvas, comprising the original in-context pair and the true label for the query, is encoded using a pre-trained `VQGAN` encoder to produce ground-truth tokens. A cross-entropy loss is computed on the empty cell to update only the parameters of the Prompt Enhancer, ϕ.

3.3.3 In-context Learning Retriever

Selecting a high-quality in-context pair that aligns well with a given query image is essential for enhancing ICL performance [31]. To improve this selection process, we advocate for leveraging rich and comprehensive visual features during retrieval. To this end, we follow the Feature-Map-Level Retrieval (FMLR) [91] strategy.

In particular, we employ a pre-trained visual encoder, different from that used in previous InMeMo works, such as the one from DINOv2 [9] to extract ℓ_2-normalized visual features for both the query image x_q and every candidate image $x \in \mathcal{S}$. Given that our downstream tasks are entirely based on image features, and considering the differing pre-training paradigms of CLIP and DINOv2, it is reasonable to adopt DINOv2, which is pre-trained on large-scale

images, as our feature extractor. Unlike methods that rely on global features, FMLR preserves the full feature map, flattens it into a one-dimensional representation, and utilizes it directly for similarity comparison.

Among all available in-context candidates in $\mathcal{S}$, the one whose visual representation is most similar to that of the query is selected as the in-context pair (x, y). Formally, the selection is given by

$$(x, y) = \underset{(x^\star, y^\star) \in \mathcal{S}}{\arg\max}\, \psi(x_{\mathrm{q}})^\top \psi(x^\star), \tag{3.3}$$

where $\psi(\cdot)$ denotes the extracted and normalized visual features.

3.3.4 Prompt Enhancer

Learnable visual prompting, initially introduced in [67], draws inspiration from prompt-based learning in NLP [16, 30, 108]. It offers an effective solution for addressing domain shifts by enabling input adaptation for downstream tasks, without requiring any fine-tuning of the base model. In our setting, we follow the formulation in [67], introducing pixel-level learnable perturbations near the image boundaries to guide the model more effectively.

In the E-InMeMo framework, the Prompt Enhancer applies such learnable prompts to the in-context image pair using different formats. These enhanced examples provide additional instructions that help the frozen model interpret the task more accurately, thereby narrowing the domain gap between the query image and the provided in-context pairs. Notably, the learnable prompt is input-agnostic; a single shared prompt is used across all in-context pairs for the same task. This allows the learned prompt to act as a task-specific identifier.

Given a retrieved in-context pair (x, y), the prompt enhancer applies a learnable perturbation t_ϕ, controlled by a parameter set ϕ, to produce the modified pair (x', y') as follows:

$$(x', y') = (x, y) + \delta t_\phi, \tag{3.4}$$

where δ defines the perturbation's strength. Unlike previous InMeMo approaches, we add a 2-grid padding type visual prompting only to the in-context pair, rather than adding it to the

in-context image and mask separately. This format provides the ICL model with more learnable parameters and fewer occlusions. The prompt t_ϕ resides in the image space and consists of learnable pixels concentrated along the image borders. All other pixels are zeroed out. These parameters are optimized through backpropagation during training.

3.3.5 Prediction

Following [10], we adopt the MAE-VQGAN model, in which pre-trained MAE [109] g generates visual tokens $\hat{\mathbf{z}}$ from $\hat{c}_\mathrm{q}$. The VQGAN [110] decoder f, again pre-trained, generates resulting image $\hat{y}_\mathrm{q}$ from $\hat{\mathbf{z}}$.

After compiling the in-context pair and the query into a canvas $\hat{c}_\mathrm{q}$, g predicts latent visual tokens $\hat{\mathbf{z}} = (\hat{z}_1, \ldots, \hat{z}_L)$, specifically,

$$\hat{z}_l = \arg\max_v g_{lv}(\hat{c}_\mathrm{q}), \tag{3.5}$$

where $\hat{z}_l \in \hat{\mathbf{z}}$ is a visual token in the vocabulary $\mathcal{V}$ at spatial position l, and g_{lv} gives the probability of $v \in \mathcal{V}$ for l. The VQGAN decoder f then generates a label image by

$$\hat{y}_\mathrm{q} = f(\hat{\mathbf{z}}). \tag{3.6}$$

We obtain the prediction for the query x_q as $\hat{y}_\mathrm{q}$.

3.3.6 Training

The only learnable parameters in E-InMeMo are the prompt t_ϕ. We train it for a specific task on $\mathcal{S}$. The loss is the same as [10], while all parameters except for t_ϕ are frozen.

We first randomly choose a pair $(x_\mathrm{q}, y_\mathrm{q})$ as query from $\mathcal{S}$. The E-InMeMo prediction process from the ICL retriever is then applied to x_q to compute $\hat{\mathbf{z}}$, but the retriever uses $\mathcal{S} \setminus \{(x_\mathrm{q}, y_\mathrm{q})\}$ instead of $\mathcal{S}$.

The label image y_q is used for training. We compile the retrieved in-context pair (x, y) and $(x_\mathrm{q}, y_\mathrm{q})$ into a canvas $c_\mathrm{gt} = (x, y, x_\mathrm{q}, y_\mathrm{q})$. The pre-trained VQGAN encoder E associated f

gives the ground-truth visual tokens $\mathbf{z}$ that reconstruct y_{q} with f, i.e.,

$$z_l = \arg\max_v E_{lv}(c_{\mathrm{gt}}), \tag{3.7}$$

where E_{lv} again is the probability of $v \in \mathcal{V}$ for position l. The loss function $\mathcal{F}$ to train our learnable prompt t_ϕ is given by

$$\mathcal{F}(\phi) = \mathbb{E}[\mathrm{CE}(g_l(\hat{c}_{\mathrm{q}}), z_l)], \tag{3.8}$$

where CE is the cross-entropy loss, $g_l(\hat{c}_{\mathrm{q}}) \in \mathbb{R}^{|\mathcal{V}|}$ is the probabilities of respective tokens in $\mathcal{V}$, and the expectation is computed over all $(x_{\mathrm{q}}, y_{\mathrm{q}}) \in \mathcal{S}$ as well as all visual tokens z_l corresponding to y_{q} (*i.e.,* over the latent visual tokens of r, represented as masked index).

3.3.7 Interpretation

Adding t_ϕ to images in a visual prompt as in Equation (3.4) translates the distribution of the prompt in a certain direction. Determining t_ϕ by Equation (3.8) will encode some ideas about the task described by $\mathcal{S}$ in ϕ, supplying complementary information that is not fully conveyed by the in-context pair (x, y). We consider that our training roughly aligns the distributions of image patches $\hat{c}_{\mathrm{q}}$ and c_{gt} in the latent space before visual token classification with smaller degrees of freedom in t_ϕ. This can be particularly effective as these distributions are inherently different due to the lack of the ground-truth label image y_{q} in the canvas. Therefore, our best expectation is that ϕ captures the distribution of y_{q} collectively to bring the distribution of prompts closer to the ground-truth prompts (containing ground-truth label y_{q}). With this, the encoder f will have better access to more plausible visual tokens that decode a label image closer to the ground-truth label.

3.4 Experiments

3.4.1 Experimental Setup

Datasets and Downstream Tasks. We follow the experimental settings of [10] to evaluate E-InMeMo. As downstream tasks, we perform foreground segmentation and single object detection. (1) Foreground segmentation aims to extract apparent objects from the query image with the in-context pair. We use the Pascal-5^i dataset [111], which is split into four-fold subsets, each containing five classes. (2) Single object detection evaluates whether a model can capture fine-grained features specified by a coarse-grained bounding box in the in-context pair. We conduct experiments on images and bounding boxes from the PASCAL VOC 2012 [112]. To align with [10], we use a subset of the dataset whose samples only contain a single object as our dataset $\mathcal{S}$.

We also verify E-InMeMo on two medical datasets for foreground segmentation. (1) Kvasir-SEG dataset [113] is a dataset for gastrointestinal polyp segmentation. It contains 1000 polyp images and their corresponding binary segmentation masks. Since there has no official split of the test set, we randomly split the whole dataset in a five-fold, to conduct the experiments. Notably, methods such as DCNet [114] have demonstrated strong performance in similar polyp segmentation tasks. (2) Skin lesion dataset [115] is a dermoscopic image benchmark challenge for diagnosis of skin cancer. It contains 900 images for training and 379 images for test. Each image contains their binary segmentation masks labeled by experts. For this specific task, approaches like PMFSNet [116] have yielded promising results.

Methods for Comparison. All experiments utilize MAE-VQGAN [10] as the pre-trained large-scale vision model. E-InMeMo is compared against SOTA methods for visual ICL. Below are the details of the comparison methods.

- Random [10] means randomly retrieves in-context pairs from the training set.
- UnsupPR [31] uses CLIP as the feature extractor to retrieve in-context pairs based on features after the classification header.

- SupPR [31] introduces contrastive learning to develop a similarity metric for in-context pair selection.
- SCS [117] proposes a stepwise context search method to adaptively search the well-matched in-context pairs based on a small yet rich candidate pool.
- Partial2Global [118] proposes a transformer-based list-wise ranker and ranking aggregator to approximately identify the global optimal in-context pair.
- Prompt-SelF [91] applies an ensemble of eight different prompt arrangements, combined using a pre-defined threshold of voting strategy.
- FMLR (CLIP) means to utilizes CLIP as the feature extractor for feature-map-level retrieval.
- FMLR (DINOv2) utilizes DINOv2 as the feature extractor for feature-map-level retrieval.

Additionally, we compare E-InMeMo with few-shot segmentation methods derived from meta-learning, such as OSLSM [111] and co-FCN [119]. For medical datasets, E-InMeMo is compared with visual ICL methods and FMLR using different feature extractors.

Implementation details. For the foreground segmentation task, E-InMeMo is trained separately on each fold of the training dataset, treating each fold as an independent task and learning a unique prompt for each one. In the case of single object detection, the model is trained using the entire training set, with in-context pairs being retrieved from the same set. During evaluation, each test image is treated as a query, and its corresponding in-context pair is selected from the training set.

All images are resized to 111×111 pixels to construct the canvas. The learnable prompting is applied around the borders of the in-context pair, occupying 15 pixels along each edge. This configuration results in a total of 27,540 trainable parameters for the prompt ϕ. The canvas layout follows the standard format from [10], where the enhanced in-context pair (x', y'), the query image x_{q}, and an empty cell r are placed at the top-left, top-right, bottom-left, and bottom-right positions, respectively. The perturbation strength δ is set to 1.

Our implementation is built with PyTorch (version 1.12.1). Training is conducted for 70 epochs using the Adam optimizer [83], starting with an initial learning rate of 15. The learning rate follows a cosine annealing schedule with warm restarts. A key strength of E-InMeMo lies in its efficiency: the model can be trained on a single NVIDIA Tesla V100-SXM2-32GB GPU using a batch size of 32. Code is publicly available at: `https://github.com/Jackieam/E-InMeMo`.

3.4.2 Comparison with State-of-the-Art

We present a comparative analysis of E-InMeMo against existing visual in-context learning (ICL) methods and meta-learning-based few-shot learning approaches, as shown in Table 3.1. The results demonstrate that E-InMeMo sets a new SOTA in both downstream tasks, with particularly strong gains in single object detection. Notably, it significantly outperforms the FMLR baseline and consistently surpasses meta-learning-based methods across nearly all folds as well as in average performance. These findings underscore the effectiveness of incorporating a learnable prompt into the visual ICL framework.

One point of comparison is prompt-SelF [91], whose performance benefits from a bagging mechanism. It applies the model across eight different canvas configurations and aggregates the results, thereby leveraging diverse perspectives from the large vision model. In contrast, E-InMeMo requires only a single inference per query, yet achieves superior results. While bagging may offer a performance boost without significant overhead, the improvements achieved by E-InMeMo remain substantial.

E-InMeMo particularly excels in single object detection, where it exceeds the prompt-SelF by an impressive 13.38 points, highlighting its ability to accurately capture fine-grained visual cues and detect small objects with high precision.

These findings illustrate the strength of our direct and efficient strategy. By refining the in-context pair through learnable prompting, E-InMeMo effectively enhances the model's task comprehension. Furthermore, the approach remains lightweight, adding only 27,540 trainable

Table 3.1: Performance of the foreground segmentation and single object detection downstream tasks. The best scores in *in-context learning* are highlighted in **bold**. **Seg.** and **Det.** stand for the foreground segmentation and single object detection tasks, respectively.

	Seg. (mIoU)					Det.
	Fold-0	**Fold-1**	**Fold-2**	**Fold-3**	**Mean**	**(mIoU)**
Meta-learning						
OSLSM [111]	33.60	55.30	40.90	33.50	40.80	-
co-FCN [119]	36.70	50.60	44.90	32.40	41.10	-
In-context learning						
Random [10]	28.66	30.21	27.81	23.55	27.56	25.45
UnsupPR [31]	34.75	35.92	32.41	31.16	33.56	26.84
SupPR [31]	37.08	38.43	34.40	32.32	35.56	28.22
SCS [117]	-	-	-	-	35.00	-
Partial2Global [118]	38.81	41.54	37.25	36.01	38.40	30.66
prompt-SelF [91]	42.48	43.34	39.76	38.50	41.02	29.83
FMLR (CLIP)	35.69	37.24	33.98	33.13	35.01	27.27
FMLR (DINOv2)	36.48	37.41	35.03	34.77	35.92	27.18
InMeMo [43]	41.65	**47.68**	42.43	40.80	43.14	43.21
E-InMeMo (Ours)	**42.82**	46.97	**43.00**	**42.83**	**43.91**	**44.22**

parameters and requiring minimal computational resources. The task-specific, shared learnable prompts not only simplify the training process but also point toward a scalable and efficient direction for future visual ICL research.

3.4.3 Results on Medical Datasets

In addition to the dataset comprising natural images, we also verify E-InMeMo on two medical datasets as a foreground segmentation task. The results are shown in Table 3.2. We observe that E-InMeMo also achieved the SOTA results on these two medical datasets, demonstrating that E-InMeMo is also effective on medical domain datasets. We found that, on the ISIC dataset, there

was a minor degradation of SupPR and FMLR (DINOv2) compared to suboptimal methods (such as UnSupPR and FMLR (CLIP)). This phenomenon demonstrates that previous methods cannot ensure obtaining better results on different datasets (such as medical datasets). But E-InMeMo also showed optimal performance on such dataset transformation.

Table 3.2: The performance of E-InMeMo on ISIC and Kvasir datasets with different diseases.

	Kvasir (mIoU)						ISIC
Method	**Fold-0**	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Mean**	**(mIoU)**
Random [10]	37.79	33.98	35.72	32.33	32.51	34.47	62.43
UnSupPR [31]	39.71	39.99	41.02	35.90	37.54	38.83	68.78
SupPR [31]	41.16	40.94	41.58	38.53	36.33	39.71	68.76
FMLR (CLIP)	41.17	40.91	40.49	39.40	37.01	39.80	70.77
FMLR (DINOv2)	45.70	42.25	44.85	43.25	39.94	43.20	70.12
E-InMeMo	**53.04**	**50.04**	**53.96**	**54.47**	**45.97**	**51.50**	**76.13**

3.4.4 Domain Shifts Analysis

In practical applications, domain shifts frequently occur, representing variations between the training dataset $\mathcal{S}$ and the target environment. These discrepancies can lead to reduced performance when models are applied beyond their original training distribution. These shifts are prevalent across datasets and are commonly used as benchmarks to evaluate the robustness of machine learning models [86].

To investigate E-InMeMo's resilience to domain shifts, we adopt the COCO dataset [120] for out-of-domain inference, following the evaluation protocol established in [31]. Specifically, we use a variant known as COCO-5^i, which aligns its category divisions with those of Pascal-5^i [31]. In this setup, referred to as COCO $\rightarrow$ Pascal, the in-context pairs are sampled from COCO-5^i, while the query images come from the Pascal-5^i validation split, consistent with the configuration in [91].

The domain shifts evaluation results are summarized in Table 3.3. Under the COCO → Pascal setting, the FMLR baseline using DINOv2 experiences a 2.32-point drop in mIoU, equivalent to a 6.46% decrease. E-InMeMo, on the other hand, achieves an mIoU of 41.42%, reflecting a 2.49-point drop (5.67% decrease). Despite the slightly larger absolute drop, E-InMeMo exhibits a smaller relative decrease, and the performance gap between the two methods under domain shifts is minimal (0.17 points). This indicates that E-InMeMo maintains greater robustness in the face of domain shifts. These findings suggest that incorporating learnable prompts

Table 3.3: Domain shifts analysis on E-InMeMo. *Pascal* → *Pascal* means in-context pairs and query images are both from PASCAL. *COCO* → *Pascal* indicates that in-context pairs are from COCO while query images are from PASCAL.

Domain Setting	Method	Fold-0	Fold-1	Fold-2	Fold-3	Mean
Pascal → *Pascal*	FMLR (DINOv2)	36.48	37.41	35.03	34.77	35.92
	E-InMeMo	42.82	46.97	43.00	42.83	**43.91**
COCO → *Pascal*	FMLR (DINOv2)	34.43	34.06	32.60	33.31	33.60
	E-InMeMo	41.28	42.26	40.97	41.18	**41.42**

into visual ICL not only improves in-domain performance but also enhances cross-domain generalization. The robustness and adaptability of E-InMeMo make it a promising approach for real-world applications where domain shifts are inevitable.

3.4.5 Further Analyses on E-InMeMo

This section further investigates the capabilities of E-InMeMo through a series of experiments, primarily focusing on the fine-grained foreground segmentation task.

Qualitative comparison. We conduct a qualitative comparison of E-InMeMo against FMLR (DINOv2), prompt-SelF [91], and the ground-truth (GT) labels across both foreground segmentation and single object detection tasks, as illustrated in Figure 3.4.

In the foreground segmentation task (Figure 3.4a), E-InMeMo produces segmentation re-

sults that closely align with the GT labels, capturing fine-grained details with high fidelity. Notably, E-InMeMo demonstrates strong robustness to various visual discrepancies. For instance, it performs well even when the in-context image is achromatic or when there is a substantial color difference between the in-context and query images. Additionally, E-InMeMo remains stable in cases involving significant foreground size variation and is capable of accurately distinguishing foreground from background in challenging scenes.

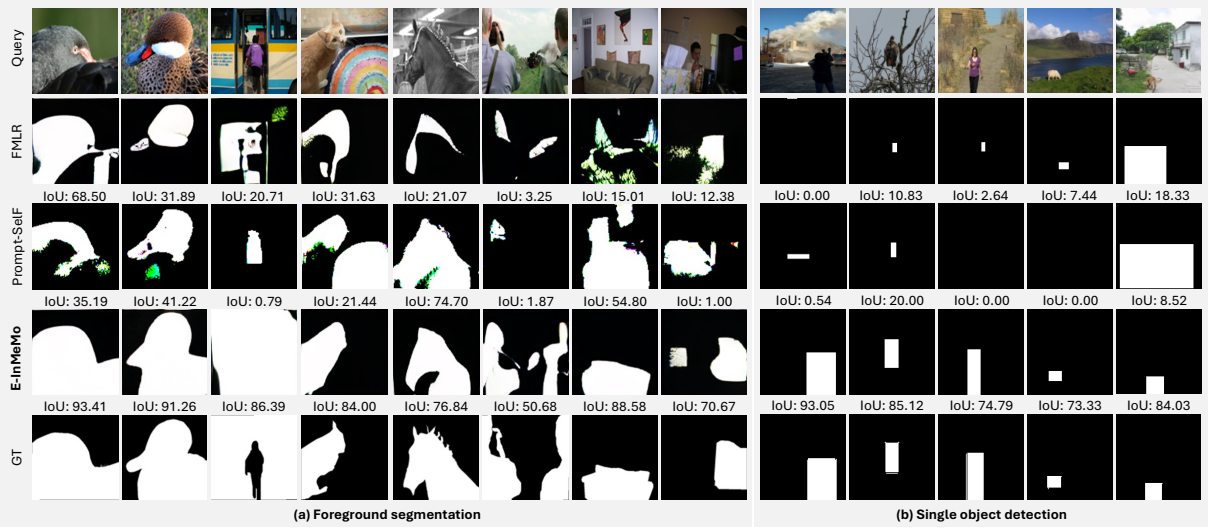


Figure 3.4: Qualitative comparisons across baseline methods, prompt-SelF, and our proposed E-InMeMo on two downstream tasks: (**a**) Foreground segmentation and (**b**) Single-object detection. For each task, the top row shows the query image. The subsequent rows (from top to bottom) present results from FMLR (DINOv2), prompt-SelF, **E-InMeMo**, and the ground-truth (GT) label, respectively. E-InMeMo consistently enables visual ICL to capture finer details and demonstrates robustness to mismatches between in-context and query images. Notably, it also appears to mitigate the negative impact of low-quality in-context pairs—an important advantage when the retriever fails to find highly similar samples.

For the single object detection task (Figure 3.4b), E-InMeMo continues to exhibit consistent attention to detail. Its performance remains unaffected by changes in color or object scale within the in-context pair. Impressively, it is able to detect and localize objects even when the

corresponding foreground in the in-context example is minimal, showcasing its adaptability and strong visual reasoning.

Figure 3.5 further showcases examples from two medical imaging datasets, Kvasir and ISIC. In these more complex and domain-specific tasks, E-InMeMo is compared with FMLR (DINOv2). The results reveal that E-InMeMo accurately segments the regions of interest, including gastrointestinal polyps and skin lesions, while FMLR consistently fails to deliver meaningful segmentation outputs. These visual results highlight the capability of E-InMeMo to generalize to medical imagery and reinforce its potential for real-world applications in specialized domains.

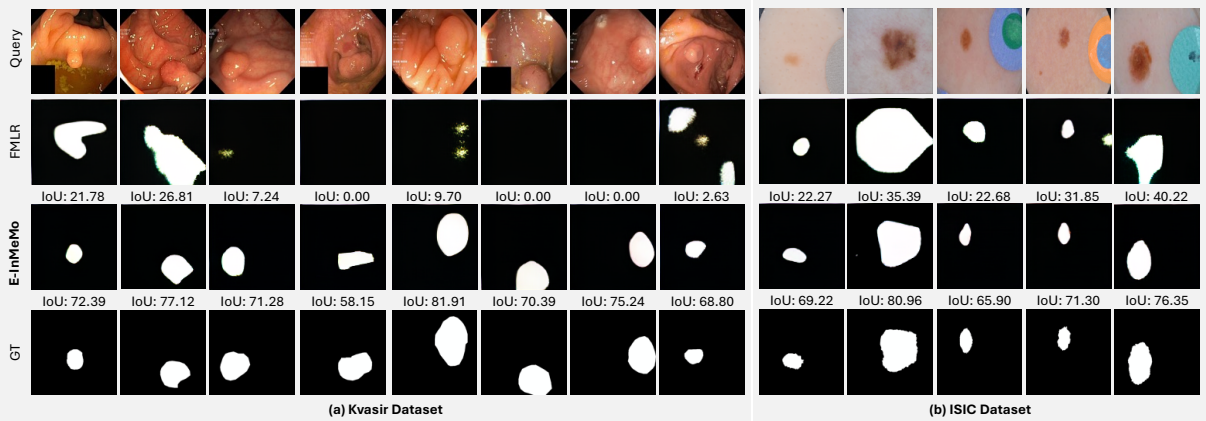


Figure 3.5: Qualitative results of FMLR (DINOv2) and our proposed E-InMeMo on two medical datasets: (**a**) Kvasir and (**b**) ISIC. For each dataset, the top row shows the query image. The following rows are arranged from top to bottom in the order of FMLR (DINOv2), **E-InMeMo**, and the ground-truth label (GT), respectively.

Where should the learnable prompt be applied? We advocate applying the learnable prompt exclusively to in-context pairs, as they play a crucial role in guiding the model during inference and are central to the effectiveness of visual ICL. To systematically investigate the impact of prompt placement, we three variants of E-InMeMo with different prompt application strategies: (1) applying the prompt only to the in-context image (`I`); (2) only to the query image (`Q`); (3) to both the in-context image and query image (`I` & `Q`); and (4) to the in-context image and

in-context label (I & L), which corresponds to the original E-InMeMo formulation.

The evaluation results are reported in Table 3.4. Across all variants, the introduction of a learnable prompt improves performance over baseline without the learnable prompt, regardless of where it is applied. Interestingly, while applying the prompt to in-context images alone (I) yields lower performance compared to other E-InMeMo variants, it still surpasses prompt-SelF. This finding reinforces that refining in-context pairs, even minimally, can significantly enhance visual ICL effectiveness.

Table 3.4: Segmentation performance for different combinations of images in the canvas to which the learnable prompt is applied. I, L, and Q denote the in-context image, in-context label image, and query image, respectively.

Combination	Fold-0	Fold-1	Fold-2	Fold-3	Mean
E-InMeMo variant					
I	42.28	44.86	41.10	43.24	42.87
Q	40.71	43.32	39.97	40.01	41.01
I & Q	39.92	44.01	39.68	40.12	40.93
I & L (E-InMeMo)	**42.82**	**46.97**	**43.00**	**42.83**	**43.91**

Surprisingly, the configuration where the prompt is added to both the in-context and query images (I & Q) performs worse than adding it to either one individually. We attribute this to the input-agnostic nature of the prompt: when the same perturbation is applied to both images, it may hinder the model's ability to distinguish task-specific cues, thus reducing its capacity to bridge the semantic gap between the query and the support examples.

Intra- and inter-class generalizability of E-InMeMo. We have demonstrated that E-InMeMo performs well on low-quality datasets and now explore its generalizability to unseen classes not included in the training dataset $\mathcal{S}$. To achieve this, we train a learnable prompt for each of the 20 classes. Specifically, let $\mathcal{S}_\omega$ denote the training subset of images and label images in Pascal-5^i for class ω. During E-InMeMo training, an image from $\mathcal{S}_\omega$ is used as a query, paired with an image and label from $\mathcal{S}_\omega$ as an in-context pair. For inter-class generalizability,

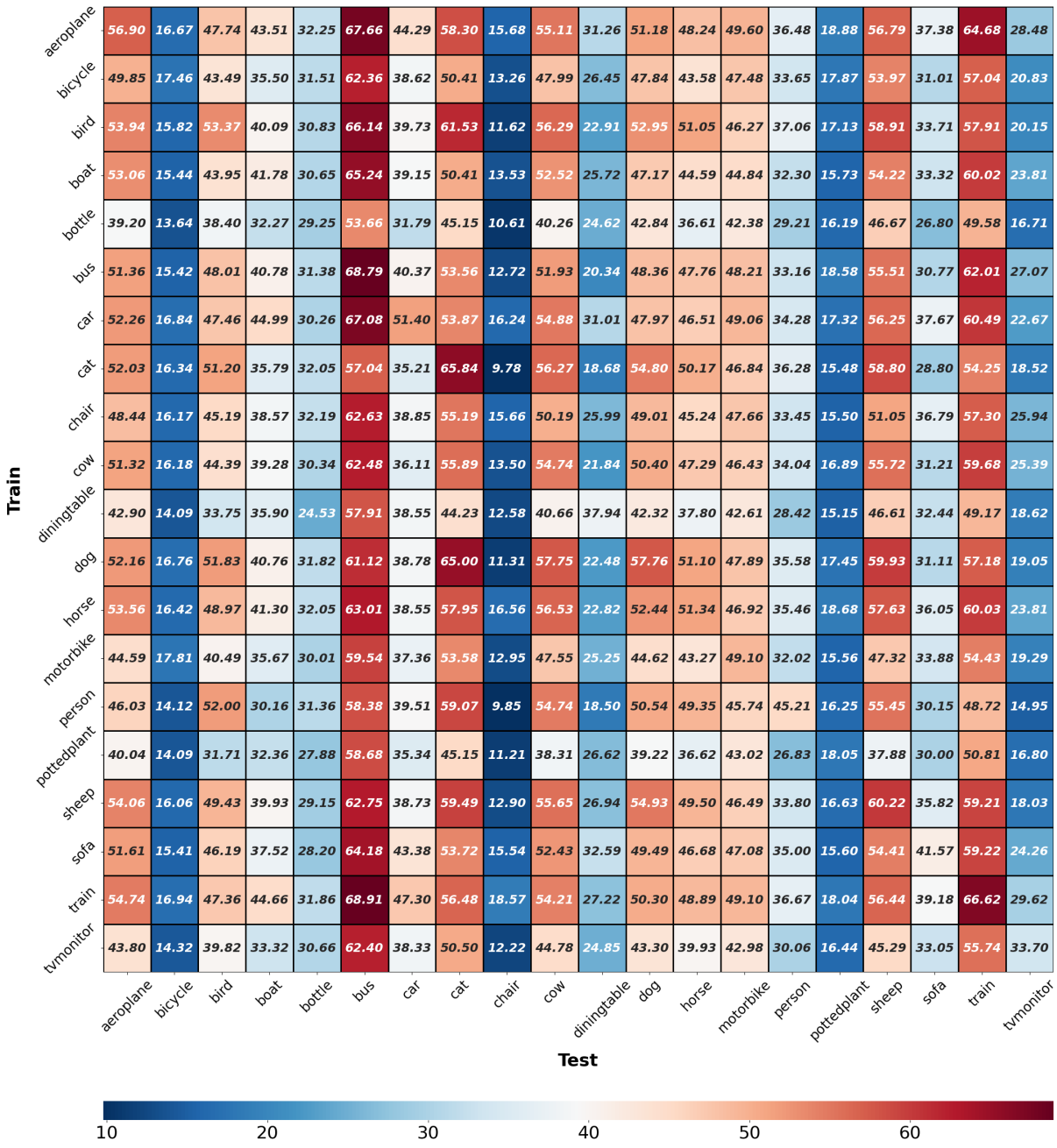


Figure 3.6: Intra- and inter-class generalization performance in mIoU. The horizontal and vertical axes are the classes used for prediction and training, respectively. The diagonal elements show intra-class performance.

predictions are made using images in the test set $\mathcal{S}_{\omega'}$ as queries, with in-context pairs retrieved from the training set $\mathcal{S}_{\omega}$, where $\omega' \neq \omega$. The full results are shown in Table 3.1. Diagonal values indicate intra-class results. The analysis reveals varying levels of difficulty across classes: (1) Prompts trained on their own class achieve good results for classes such as `bus`, `cat`, and `train`. (2) Prompts trained on their own class perform poorly for classes such as `bicycle`, `chair`, and `pottedplant`.

We also observed that, for both challenging and general classes, the overall performance of prompts adapted from different classes aligns with intra-class performance (best viewed vertically in Figure 3.6). For instance, `bus`, `cat`, `sheep`, and `train` are the most *generalizable* classes, with prompts trained on different classes achieving high accuracy (IoU close to 50%). In contrast, classes such as `bicycle`, `chair`, and `pottedplant` are the least generalizable, performing poorly across all prompts. These results suggest that, for challenging classes, learned prompts struggle to deliver good performance, whereas for general classes, learned prompts consistently achieve strong results.

In our inter-class analysis (best viewed horizontally in Figure 3.6), we observe that the effectiveness of prompts trained on different classes varies based on the inherent difficulty of each target class. Notably, certain patterns of generalizability emerge across related categories. Within the `transportation` super-class, comprising classes such as `aeroplane`, `train`, and `car`, prompts exhibit strong cross-class generalization. For example, prompts trained on `aeroplane` generalize well to `train` and `bus`, and similarly for `car`-`train` and `motorbike`-`train` pairs.

In contrast, prompts from `transportation` classes tend to generalize poorly to the `animal` category (e.g., `dog`, `horse`). Within the `animal` group, most classes show good mutual generalizability, with the exception of `cow`, which consistently underperforms. However, prompts from `animal` classes also struggle to generalize to `transportation` classes.

There are a few interesting exceptions. For instance, the `aeroplane` class generalizes surprisingly well to `cat`, and `bird` performs well when applied to `bus`. These anomalies may be attributed to class-specific segmentation properties, for example, `cat` and `bus` may

inherently be easier to segment, regardless of the prompt source.

The mean mIoU score over all pairs of 20 classes is 39.24%. This score surpasses most methods in Table 3.1 but represents a significant drop compared to E-InMeMo's mean score over all folds. This finding underscores the importance of tuning the learnable prompt for target tasks.

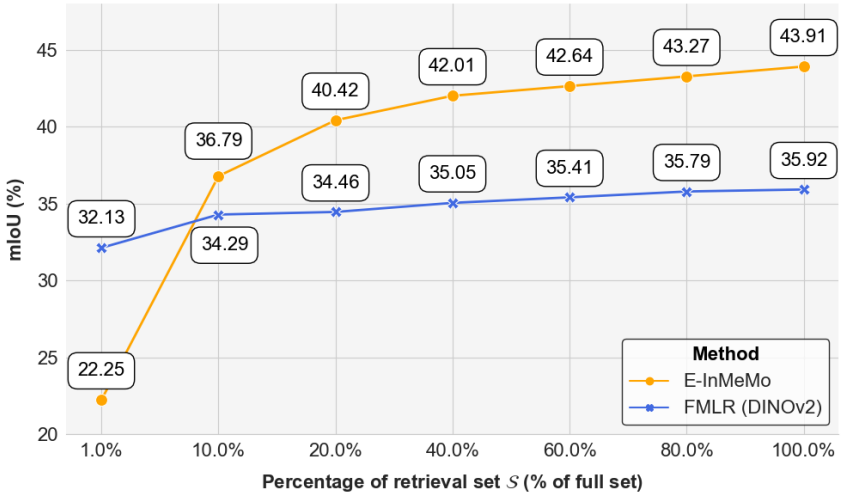


Figure 3.7: The performance of E-InMeMo and FMLR (DINOv2) in mIoU of each fold for the size of retrieval set $\mathcal{S}$. We annotate the scores in the figure.

Is E-InMeMo sensitive to the size of retrieval dataset? Following [31], we wonder whether E-InMeMo is sensitive to the size of the retrieval set $\mathcal{S}$ or not. So we use seven different subsets (1%, 10%, 20%, 40%, 60%, 80%, and 100%) on each fold as retrieval sets to be used for retrieval and as training sets for E-InMeMo training. The results are shown in Figure 3.7. It is obvious that both FMLR (DINOv2) and E-InMeMo benefit from larger retrieval set sizes. There is a clear trend in the growth of E-InMeMo; especially when going from 1.0% to 10.0%, there is a significant increase in mIoU. E-InMeMo is more sensitive to the data, and when the retrieval set is increased from 1% to 100%, its mIoU improves by 21.66 points. FMLR (DINOv2), on the other hand, is not as sensitive to the size of the retrieval set and improves by only 3.79 points. It should be noted that E-InMeMo needs only 10% of the retrieval set to achieve significantly better results than FMLR (DINOv2) at 100% retrieval set with parameter-efficient fine-tuning

which, side by side, emphasizes that E-InMeMo is a data-efficient method.

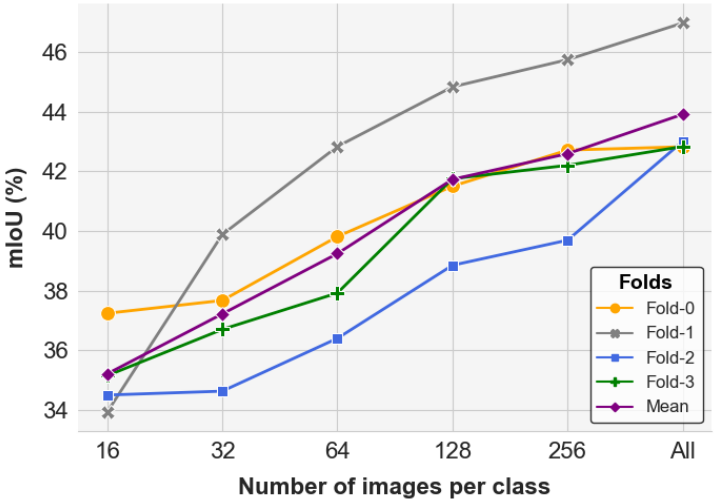


Figure 3.8: The performance of E-InMeMo in mIoU of each fold for the number of images per class in $\mathcal{S}$. All means to use all images in the training set.

Is E-InMeMo sensitive to dataset size? To better understand how the performance of E-InMeMo scales with the size of the dataset $\mathcal{S}$, we investigate its sensitivity to varying data volumes. Given the lightweight and efficient nature of the learnable prompt, we systematically varied the dataset size by randomly selecting 16, 32, 64, 128, and 256 images per class for each fold. The performance trends are illustrated in Figure 3.8.

The experimental results indicate that E-InMeMo outperforms the baseline mIoU score of 35.92% once the dataset includes at least 64 images per class, achieving 37.22% at that point. Across all folds, performance generally improves as more data is provided. For instance, in Fold-0, E-InMeMo reaches 37.24% with only 16 images per class and shows rapid gains starting from 32 images, eventually saturating at 256. Both Fold-1 and Fold-2 demonstrate steady and significant improvements with increasing data volume.

In Fold-3, a more complex fold, a notable jump in performance is observed between 64 and 128 images per class. Beyond this, the improvement plateaus, with only marginal gains when utilizing the full dataset.

Overall, these results suggest that E-InMeMo requires fewer training examples for simpler

folds, yet benefits from larger datasets in more complex scenarios. This data efficiency makes E-InMeMo particularly suitable for tasks where labeled data is limited, while still being capable of leveraging larger datasets when available to further boost performance.

Table 3.5: The mIoU scores for varying padding sizes of prompt enhancer t_ϕ. The Para. represents the number of tunable parameters. The highest score in each fold is marked in **bold**.

Padding Size	Para.	Fold-0	Fold-1	Fold-2	Fold-3	Mean
5	9,780	41.31	47.12	41.37	41.14	42.73
10	18,960	**43.59**	**47.86**	41.46	42.65	43.89
15 (E-InMeMo)	27,540	42.82	46.97	**43.00**	**42.83**	**43.91**
20	35,520	39.23	42.81	37.77	41.06	40.22
25	42,900	42.22	45.96	37.96	39.72	41.47
30	49,680	31.53	40.62	36.10	39.26	36.88

Effect of Padding Size in the Prompt Enhancer t_ϕ. In our implementation, the padding size for E-InMeMo's learnable prompt enhancer t_ϕ is set to 15 pixels. To investigate the influence of different padding sizes on model performance, we conducted a comprehensive evaluation, with the results summarized in Table 3.5.

E-InMeMo consistently outperformed all baseline methods across different padding sizes in terms of Mean performance. As the padding size increased from 5 to 15, we observed a steady improvement, suggesting that a moderate spatial footprint for the prompt allows the model to better guide the in-context pair without excessive interference.

However, further increasing the padding beyond 15 pixels led to diminishing returns. Specifically, increasing the padding size to 20 resulted in a noticeable drop in performance, from 43.91% to 40.22% on Mean. Although a small gain was observed when moving from 20 to 25 pixels, performance remained below the optimal. When the padding was increased to 30, the performance continued to decline.

Interestingly, at a padding size of 10, the model achieved the best results on Fold-0 (43.59%) and Fold-1 (47.86%), yet overall Mean performance was slightly lower than that of padding size

15. This suggests that, while smaller prompts can be beneficial in specific scenarios, they may lack the generalization capacity required across more diverse or complex folds. We hypothesize that larger padding sizes (e.g., larger than 20) cause the learnable prompt to occupy too much of the original image, potentially overwriting critical visual information from the in-context pair. This over-perturbation compromises the prompt's guiding ability and leads to a drop in performance.

In summary, our findings indicate that a padding size of 15 offers the best trade-off between expressive capacity and preservation of the original visual content, making it the empirically optimal setting for E-InMeMo. Expanding the padding further introduces unnecessary parameters and can degrade performance.

3.5 Discussion

We proposed E-InMeMo, a parameter-efficient method for applying vision foundation models to specific tasks and domains. E-InMeMo has broad applicability in fields such as medicine and environmental science, where ease of adaptation and high accuracy are essential. The basic performance of visual in-context learning (ICL) can be further enhanced depending on the strength of the underlying vision foundation models [10, 31]. Leveraging powerful models and integrating E-InMeMo with data- and parameter-efficient strategies could be a promising direction for real-world deployment.

Limitations. Despite its strengths, E-InMeMo requires a minimum of 32 images per class to achieve competitive performance relative to the baseline. Furthermore, the learned prompt is task-specific, a prompt trained for one class or task does not generalize well to others. Therefore, designing dedicated prompts for each target task remains essential to maximizing performance.

3.6 Summary

E-InMeMo achieves SOTA performance on two downstream tasks by introducing a learnable prompt to the in-context pairs, a lightweight yet powerful approach that enhances the effectiveness of visual in-context learning (ICL). This learnable prompt improves the model's ability to recover fine-grained details in predictions and mitigates the negative impact of low-quality or semantically distant in-context pairs. Furthermore, our experiments demonstrate that E-InMeMo exhibits robustness to domain shifts, such as transferring from the Pascal dataset to COCO.

Chapter 4

VASCAR: Training-Free Visual Self-Correction for Layout Generation

4.1 Overview

Content-aware layout generation is a kind of layout generation task [121, 122] where layout elements are generated on the basis of a given visual content (i.e., canvas). It has emerged as a pivotal area of research with applications covering various tasks such as poster creation [123, 124] and magazine layouts [125, 126]. Usually, designers practice iterative processes [1] and heuristics based on guidelines [2] to achieve visually appealing layouts (Figure 4.1(a)), requiring significant expertise. However, emulating this process automatically is challenging due to the scarcity of high-quality, annotated datasets, which are difficult and costly to obtain.

Previous studies have attempted to automate content-aware layout generation by employing generative models, such as generative adversarial networks (GANs) [124, 127], Transformer-based models [3], and diffusion-based models [128, 129], to produce layouts based on visual content. While these methods have achieved remarkable success, they often lack the iterative refinement process like human designers. Some studies [130, 131] try to address this issue by introducing mechanism for iterative layout correction, yet these efforts still fall short due to the is-

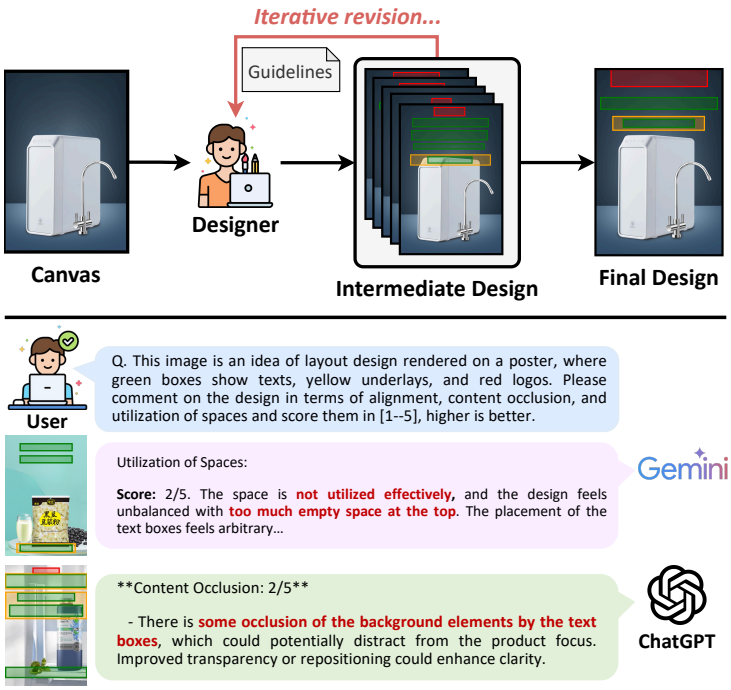


Figure 4.1: The designers' workflow follows an iterative revision process [1] and heuristics based on guidelines [2]. The conversation highlights the ability of VLMs to evaluate and identify design issues on *rendered* problematic layouts, and provide valuable feedback. This capability is leveraged *appropriately* to improve content-aware layout generation, like the workflow of designers.

sue of limited data availability. Consequently, the layouts generated by these models frequently require manual adjustments and fine-tuning by the designers to meet aesthetic standards. This highlights a significant gap in existing methodologies, where the ability to iteratively improve and refine layouts autonomously remains underdeveloped.

Recent advancements in large language models (LLMs) have demonstrated their remarkable capacity to generate various forms of structured content, including layout designs, due to their vast knowledge base, which encompasses not only text but also design principles. By fine-tuning LLMs with extensive training data and computational resources, it has become possible to leverage their inherent understanding for layout generation [132, 133]. Notably, proprietary

LLMs have recently achieved impressive performance in this domain without additional training, using in-context learning (ICL) [30] to generate layouts from only a few examples [134]. These models have shown the ability to produce high-quality layouts, even when provided only with textual descriptions of layout elements. However, since their gradients and parameters are inaccessible, traditional tuning methods are infeasible. Furthermore, two key challenges remain in applying such methods to content-aware layout generation: 1) how to efficiently leverage visual content to enhance performance is unresolved, and 2) the aesthetic quality of generated layouts is not necessarily pleasing in subjective and objective evaluation.

To address these challenges, efficiently integrating multi-modal capabilities into layout generation is essential. However, transitioning the LLM's text-centric solution to environments that incorporate multiple modalities introduces new challenges in enhancing, evaluating, and validating multi-modal content, like sequences of interleaved images and text [135]. As shown in Figure 4.1, inspired by the ability of large vision-language models (VLMs) that can assess the quality of graphic design [136], and the iterative design process [1] following guidelines and heuristics [137, 138] commonly adopted by designers to create appealing layouts, as the final work of this thesis exploring training-free interaction, we propose ***V**isual-**A**ware **S**elf-**C**orrection L**A**yout Gene**R**ation* (**VASCAR**). VASCAR is a multi-modal *training-free* framework for iteratively self-improving content-aware layout generation, designed to *mimic designers' workflow* (Figure 4.1(a)).

Specifically, we propose using bounding box layouts rendered on canvas as visual prompting, since the findings show GPT-4V's ability to understand a variety of markers [139, 140]. Given a canvas as a query and a few multi-modal ICL examples, we prompt the VLM to generate multiple layout candidates in a *parsable* HTML format. To mimic the design process, inspired by [141], we introduce a multi-modal self-correction method. The generated layouts are parsed and rendered on the query canvas as visual content. Additionally, we propose a layout scorer to assign unified scores as evaluation results and a suggester to provide recommendations for refining the generated layouts, mimicking established guidelines and heuristics. The optimized multi-modal prompt is then fed back to VLM for another iteration, as self-refinement has

been shown to improve performance in NLP tasks [135]. This process enables VLMs to refine and enhance the layout design iteratively, much like a designer practicing to create aesthetical designs.

In experiments, we confirmed the effectiveness of VASCAR using the PKU and CGL poster-layout datasets [124, 127]. On the relatively small PKU with 10k-order training samples, VASCAR outperformed all of the existing LLM-based and generative-model-based methods across most constraint-based metrics such as occlusion, readability, and distributional metrics, layout FID [142]. In larger CGL, where layout-specific generative models enjoy their full modeling abilities, VASCAR still outperformed them in most of the constraint-based metrics and marked competitive FID without training.

Contributions. We leverage VLMs for content-aware layout generation in a *training-free* manner for the first time. We propose a simple but effective method using bounding-box layouts rendered on canvas as visual prompting, serving as visual content for VLMs. To mimic designers' workflow, we propose a layout scorer and a suggester to iteratively prompt VLMs. Extensive experiments and user study show that VASCAR achieves the state-of-the-art (SOTA) content-aware layout generation results. The generalizability of VASCAR across VLMs (e.g., GPT-4o and Gemini) demonstrates its versatility. Our findings on VLMs' usability for content-aware layout generation serve as a step-stone for future research in advanced graphic design with VLMs.

4.2 Related Work

4.2.1 Layout-Specific Generative Models.

The exploration of layout generation [121,122] began in a content-agnostic setting, initially employing optimization-based methods guided by design principles [143, 144]. With the advent of deep learning, methods based on generative adversarial networks (GANs) [145, 146] and variational auto-encoders (VAEs) [147, 148] were proposed, marking a significant shift in the

field. Notably, optimizing the latent variables of GANs to generate layouts, an approach known as CLG-LO [142], has demonstrated a certain efficacy in aesthetic constraints. Furthermore, LACE [4] demonstrated performance improvements by employing a post-processing method that directly optimizes the constraints. Since then, Transformer [149]-based models [150, 151], and more recently, diffusion-based [131, 152, 153] and flow-based [154] models have been introduced to enhance the generative capabilities in content-agnostic scenarios.

Following the research in content-agnostic settings, content-aware layout generation, which is more readily applicable in business contexts, has emerged, with GAN-based [124, 127, 155] and diffusion-based [128, 129] methods being proposed. These models are trained from scratch, but acquiring design data is often more costly in terms of human and financial resources, leading to a limited amount of data compared to other domains. As recent models become increasingly data-hungry, the scarcity of training data remains a major challenge in this field.

4.2.2 Layout Generation by LLMs.

Recent studies pointed out that LLMs have ability to generate layouts in the form of structure-description language such as HTML [132, 134]. This might be surprising because LLMs do not have explicit handling mechanisms for visual information, but it is possible that LLMs have learned to solve layout as a programming task [156] from design-related sample codes of HTML contained in the training corpora. This prior knowledge makes LLM-based layout generation promising, especially in data-scarce scenarios. LayoutNUWA [132] is one of the pioneering work in this line by formulating layout generation as a code-completion task using Code Llama [157] language model. LayoutPromter [134] and LayoutGPT [158] similarly utilized GPT-3 [30] and showed that it can adapt to layout generation in a training-free manner by using ICL. MuLCO [159] tackled multi-page consistent layout generation with GPT. Although extending them with visual modality is a natural direction, we are aware of only one existing study at present: PoseterLlama [133] carefully fine-tuned a VLM, i.e., MiniGPT-4 [160]. However, how to exploit proprietary VLMs that we can not finetune for layout generation is an open

problem, and VASCAR is the first work to tackle this.

4.2.3 Self-Correction in LLMs

For exploiting full potentials of LLMs, various prompting techniques are examined rather than simply breaking target tasks down into questions and answers, which established a research area of prompt engineering [161]. Self-correction [162] is one of the promising directions among them, where LLMs are tasked to correct mistakes in their own outputs with or without external feedback. For example, feedback of search results from external corpora is used in text-based question answering tasks [163], execution results are used as feedback in program synthesis [164], and object detector is used for text-to-image generation [165]. Interestingly, even without access to external information, LLMs can correct their own errors via prompting. This is called intrinsic self-correction [166], although its efficacy is controversial [167, 168] in the language processing area. In this context, VASCAR exploits two feedbacks for self-correction: external feedback from automatic metrics and intrinsic multi-modal feedback from rendered layout images, where the model's own textual output is fed back in the form of visual input.

4.3 Method

Figure 4.2 shows an overview of VASCAR. To achieve iterative self-corrective layout generation with a VLM, our pipeline is based of the novel *visual-aware self correction*, which consists our key contribution. In each step, VLM-based layout generation and refinement are performed. VASCAR consists of *layout generator* to generate the initial layouts, *prompt optimizer* to evaluate the layouts based on automatic criteria and construct the next-step prompts, and *ICL retriever* to identify multiple exemplars that serve as potential design constraints, each of whose details are described in the following sections.

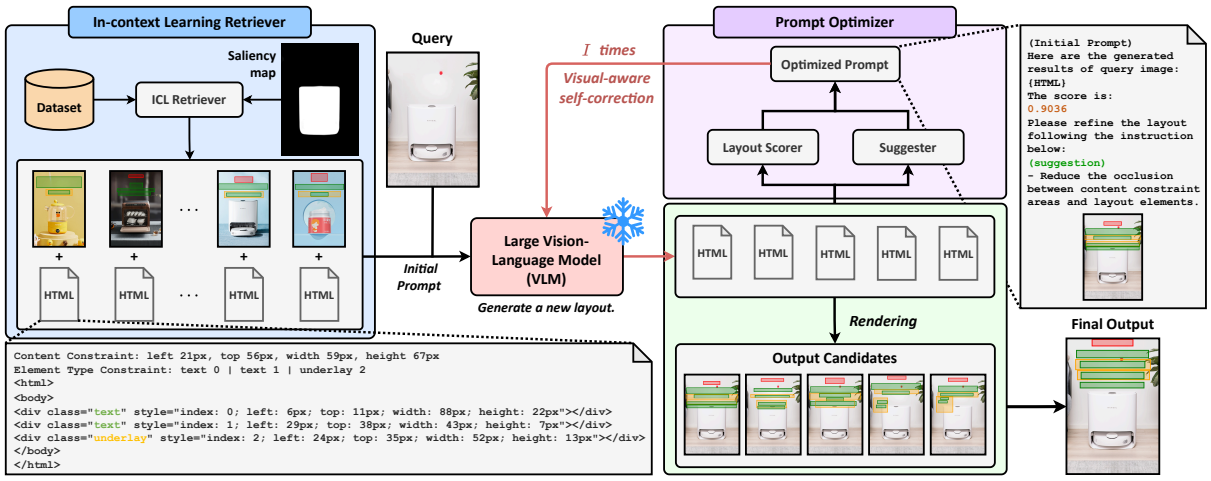


Figure 4.2: An overview of VASCAR. First, *ICL retriever* uses the saliency map of the query image to retrieve several ICL examples from the dataset. The retrieved samples are transformed to rendered images along with the text layout in HTML format. We input all the ICL examples and the query into a frozen VLM to generate the layout in HTML format. The VLM is queried iteratively with optimized prompts by *layout scorer* and *suggester*. After I iterations, VASCAR return the candidate with the highest score.

4.3.1 Task Definition

Let x_{q} denote the input canvas on which we generate a layout, and $\mathcal{D} = \{(x, y)\}$ the dataset of pairs of a canvas x and corresponding annotated *layout* y, where $|\mathcal{D}| = N$. The canvas x may be an image of a product, an event, etc., but without any visual layout elements.

A layout is a set of annotated visual elements in the materials, i.e., $y = \{\boldsymbol{e}\}$ where $|y| = K$ and each visual element $\boldsymbol{e} = (c, l, t, w, h)$ is represented by its *element type* c to specify the type of the visual element (e.g., logo, text, and underlay) and the bounding box specified by the left-top-width-height coordinates. That is, the arrangement of the visual elements in professionally designed material is encoded in y.[1] VASCAR generates a layout y_{q} for x_{q} with referring to a

[1]Note that y does not provide any information on the content of each visual element $\boldsymbol{e} \in y$. For example, if $\boldsymbol{e}$ is a text block, the text should be provided to fully re-generate the original material. We omit the content as VASCAR does not depend on it.

subset of $\mathcal{D}$ identified by the ICL retriever.

4.3.2 Visual-Aware Self-Correction

Visual-aware self-correction refers to our overall pipeline that first generates layouts on the query canvas, and refines the layouts iteratively with visual feedback. First, we introduce a unified notation for prompt-based generation and refinement with an LLM or VLM by $Y_{\mathrm{q}} = G(x_{\mathrm{q}}, Y; p)$, where $G(\cdot)$ represents the generator, x_{q} denotes the input canvas, Y denotes the current state of layouts under refinement if provided, and p denotes a prompt text that instruct the model to generate or refine layouts. The output Y_{q} is a set of layouts each of whose elements is denoted by y_{q}, since a generator may output multiple candidates of layouts per call. The initial layout generation can be denoted as

$$Y_{\mathrm{q}}^{0} = G(x_{\mathrm{q}}, \emptyset; p_0), \tag{4.1}$$

where p_0 is the default prompt that instructs the model to generate layouts from scratch. To indicate the absence of the previous-step layouts, we explicitly use an empty set $\emptyset$.

To iteratively refine the layouts with self-feedback, the same LLM or VLM used for generation can be employed by simply switching the prompt to a refinement mode. We construct refinement prompts by modifying the default prompt with additional instructions and the previous states of the generated layouts. This refinement step can be denoted by

$$Y_{\mathrm{q}}^{i} = G(x_{\mathrm{q}}, Y_{\mathrm{q}}^{i-1}; p(Y_{\mathrm{q}}^{i-1})), \tag{4.2}$$

where Y_{q}^{i} denotes layouts generated in the i-th iteration. The prompt $p(Y_{\mathrm{q}}^{i-1})$ is a refinement prompt that is conditioned by the previous state Y_{q}^{i-1}.

In constructing the function G with a VLM, we need to leverage its ability to access layout aesthetics by directly feeding a rough proxy of the resulting design with visual elements being given as their bounding boxes [136], as shown in Figure 4.1. We propose to feed these proxies, referred to as *rendered images*, to the VLM to provide richer cues for better layouts. A rendered image is generated by placing bounding boxes with different colors for different element types,

denoted by $\boldsymbol{z}(x, y)$. Most VLMs accept a text prompt combined with multiple images, and we can concretize the function G using a VLM with a capacity for visual feedback by

$$G(x_{\mathrm{q}}, Y; p) = \mathrm{VLM}(\{\boldsymbol{z}(x_{\mathrm{q}}, y_{\mathrm{q}}) | y_{\mathrm{q}} \in Y\}, p), \tag{4.3}$$

where $\mathrm{VLM}(V, t)$ denotes a VLM viewed as a function that maps a pair of image set V and prompt text t on the output layout candidates.

Positive reinforcement in candidate-set feedback has been shown to be beneficial [141] by keeping high-score examples for better results. Therefore, we select the top-5 layout from candidate pool using a scoring function $v(y_{\mathrm{q}})$ from Y_{q} and discard the others to create the reference samples for the multi-modal optimized prompt. The initial prompt is also added into the multi-modal optimized prompt, as the VLM needs to generate a parsable HTML layout. We perform self-correction for I iterations by modifying the samples in the multi-modal optimized prompt, mimicking a designer's iterative process, and select the layout with the highest score among Y_{q}^{I} as the final output.

4.3.3 Prompt Strategy for Layout Generator

Even for state-of-the-art VLMs, generating properly formatted layouts remains challenging without exemplification. To address this, we construct our layout generator by a frozen VLM enhanced with ICL [43, 97, 169, 170], incorporating a small subset of the training set into the prompts. Given an input x_{q}, we extract a small subset from the training set, denoted as $\mathcal{S}(x_{\mathrm{q}})$, where the subset size is M. These examples serve as ICL exemplars, allowing the layout generator to generate a layout y_{q} w.r.t. x_{q} and $\mathcal{S}(x_{\mathrm{q}})$. The prompt p_0 consists of (i) instructions for generating a layout, (ii) the input image x_{q}, (iii) its saliency map s_{q}, (iv) rendered image $\boldsymbol{z}$ of each ICL example in $\mathcal{S}(s_{\mathrm{q}})$, (v) saliency map s of each ICL example, and (vi) the corresponding layout y. Following [134], the saliency map is represented as a serialized bounding box. Taking s_{q} as an example, it can be represented by a serialized bounding box $(l_{\mathrm{q}}, t_{\mathrm{q}}, w_{\mathrm{q}}, h_{\mathrm{q}})$, and

is encoded into a textual format through the transformation E_{s}, i.e.,

$$E_{\mathrm{s}}(s_{\mathrm{q}}) = \text{“}\texttt{left } l_{\mathrm{q}} \texttt{ px, top } t_{\mathrm{q}} \texttt{ px,} \\ \texttt{width } w_{\mathrm{q}} \texttt{ px, height } h_{\mathrm{q}} \texttt{ px.”} \tag{4.4}$$

With this default initial prompt p_0, the VLM generates multiple layouts Y_{q}^0, where different layouts can be generated by properly setting the temperature of the VLM. Additionally, we reuse the rendered images defined in Equation 4.3 for ICL samples to exploit VLMs' multi-modal ability. All rendered images from $\mathcal{S}(x_{\mathrm{q}})$ and the input image x_{q} are fed into the VLM alongside the rendered images for visual feedback. In the self-correction steps, refinement prompts $p(Y_{\mathrm{q}})$ follow the same structure as the initial prompt but differ in that they include the previous layout candidates, combined with textual *refinement suggestion* as described in Section 4.3.4, in addition to the ICL examples. The examples of real prompts can be found in Appendix.

In-Context Learning Retriever. Applying ICL to the visual domain requires selecting ICL samples that are relevant to the query image [10, 31, 43]. We use the same retrieval rule as in previous work [134], selecting samples with saliency maps [171, 172] similar to that of the query. To efficiently reuse the saliency maps and images in Equation 4.4 and Equation 4.2, we extend the dataset $\mathcal{D}$ to include a saliency map for each pair, i.e., $\mathcal{D}' = \{(x, y, s) \mid (x, y) \in \mathcal{D}\}$.

4.3.4 Multi-modal Prompt Optimizer

VLMs' capability to access the suitability and aesthetics of generated layout is strong but still limited compared to fine-tuned models [133]. Generating multiple candidate layouts with ICL examples and choosing the best one can still be unsatisfactory [134]. Inspired by [141], and designers who iteratively create appropriate layouts for specific contexts and refine them based on guidelines [1, 2], we propose a multi-modal prompt optimizer. To achieve this, we introduce a *layout scorer* to evaluate the generated layout and a *suggester* to guide the VLM in making effective adjustments.

The layout scorer uses multiple layout evaluation metrics (detailed in Section 4.4.1). Each

metric is normalized to [0, 1] to fuse them, where the larger value means better in terms of the criterion. Let $\mathcal{M}$ be the set of criteria, and $f_m(y_\mathrm{q})$ denote the normalized score of layout y_q for criterion $m \in \mathcal{M}$. The fused score is given by:

$$v(y_\mathrm{q}) = \sum_{m \in \mathcal{M}} \lambda_m \, f_m(y_\mathrm{q}), \tag{4.5}$$

where λ_m is the empirically determined weight for criterion m in the weight vector $\boldsymbol{\lambda}$.

The fused score $v(y_\mathrm{q})$ gives the layout generator ideas about how good y_q is, but it does not tell which aspects are good and which are not. Therefore, we propose a simple and effective suggester. The suggester uses individual evaluation metric f_m. If it falls below a preset threshold, an additional instruction text is added, indicating the adjustment direction (e.g., "*Reduce the overlap.*"). The threshold θ_m is set to the average of each evaluation metric across the ICL examples in $\mathcal{S}(s_\mathrm{q})$, i.e.,

$$\theta_m = \frac{1}{|\mathcal{S}(x_\mathrm{q})|} \sum_{y \in \mathcal{S}(x_\mathrm{q})} f_m(y). \tag{4.6}$$

The additional prompts from the layout scorer and suggester are then appended to the instruction in the refinement prompt $p(Y_\mathrm{q}^i)$.

4.4 Experiments

4.4.1 Experimental Setup

Datasets. We conducted our experiments on two open-source datasets: PKU [124] and CGL [127], both of which contain e-commerce posters featuring shopping product images. The PKU dataset includes three element types: *logo*, *text*, and *underlay*. CGL provides an extra element type, *embellishment*. The PKU dataset consists of 9,974 annotated posters and 905 unannotated posters (image-only). Meanwhile, CGL offers 60,548 annotated posters and 1,000 unannotated posters. To create image-layout pairs, following previous studies [124, 127], we used the state-of-the-art inpainting model [173] to generate images devoid of any visual elements from the posters following [3]. Since neither dataset includes an official test split, we

adhered to the dataset split and experimental framework in RALF [3]. RALF divided the data into 7,735/1,000/1,000 posters for training/validation/testing in PKU, and 48,544/6,002/6,002 for CGL. For evaluation, we retrieved ICL examples solely from the training set and evaluated on both annotated and unannotated posters.

Comparisons. We compare VASCAR with task-specific methods, such as CGL-GAN [127], DS-GAN [124], ICVT [174], LayoutDM [152], Autoreg and its extension RALF [3]; with LLM-based layout generation, LayoutPrompter [134] and *training-needed* multi-modal model PosterLlama [133];[2] and with Real Data (with ground-truth layout) for reference. We reproduced LayoutPrompter and PosterLlama's results due to differences in experimental setup (e.g., train/test splits).

Evaluation Metrics. We adopt the following metrics from two perspectives as in previous studies [3, 124].

- **Content metrics** consider the harmony between the generated layout y and the original image x. *Occlusion* (Occ ↓) calculates the overlap between the main visual (specified by s_{q}) and all visual elements. *Unreadability* (Rea ↓) evaluates the spatial gradients of x within all visual elements of the element type of *text*.

- **Graphic metrics** evaluate layout quality without factoring in the visual content on the canvas. *Overlay* (Ove ↓) calculates the average IoU between all pairs of elements, excluding ones of element type *underlay*. *Non-alignment* (Align ↓) assesses the degree of spatial misalignment between elements [146]. *Underlay Effectiveness* (Und ↑) evaluates the ratio of valid *underlay* elements (i.e., the other elements are completely encompassed by the *underlay* element) to total underlay elements. *Fréchet Inception Distance* (FID ↓) measures the similarity between the distributions of real and generated layouts using a pre-trained model [142].

Implementation Details. `gemini-1.5-flash` and `gpt-4o` are used as the VLMs. We

[2]PosterLlama needed re-training in our split, and we only could reproduce it in CGL, for which configuration files and preprocessed data were provided in the official codebase.

set the number M of ICL examples to 10. The number of candidates (i.e., $|Y_q|$) is set to 5. The number I of self-correction is set to 15 and 5 for Gemini and GPT-4o, respectively. Since GPT-3 [30]'s `text-davinci-003` used in LayoutPrompter [134] is no longer available, we instead use `gpt-4o` for reproduction. We set the evaluation criteria for the prompt optimizer as $\mathcal{M} = \{\text{Occ}, \text{Rea}, \text{Ove}, \text{Align}, \text{Und}\}$,[3] and the corresponding weights $\boldsymbol{\lambda} = [0.4, 0.4, 0.1, 0, 0.1]$. Specifically, the weight for Align, λ_{Align}, is set to 0 as the outputs were almost immune to it. The temperature for Gemini and GPT-4o is set to 1.4 and 0.7, respectively.

Table 4.1: Unconstrained generation results on the PKU and CGL test split. Real Data is the test set with ground truth. † shows our reproduced results based on the RALF split [3]. Best and second-best results are **bold** and with underline, respectively.

Method	*Training-free*	PKU						CGL					
		Content		Graphic				Content		Graphic			
		Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Real Data	-	0.112	0.0102	0.0038	0.99	0.0009	1.58	0.125	0.0170	0.0024	0.98	0.0002	0.79
CGL-GAN [127]	✗	0.138	0.0164	0.0031	0.41	0.0740	34.51	0.157	0.0237	0.0032	0.29	0.1610	66.75
DS-GAN [124]	✗	0.142	0.0169	0.0035	0.63	0.0270	11.80	0.141	0.0229	0.0026	0.45	0.0570	41.57
ICVT [174]	✗	0.146	0.0185	0.0023	0.49	0.3180	39.13	**0.124**	0.0205	0.0032	0.42	0.3100	65.34
LayoutDM [152]	✗	0.150	0.0192	0.0030	0.41	0.1900	27.09	0.127	0.0192	0.0024	0.82	0.0200	2.36
Autoreg [3]	✗	0.134	0.0164	0.0019	0.43	0.0190	13.59	0.125	0.0190	0.0023	0.92	0.0110	2.89
RALF [3]	✗	0.119	0.0128	0.0027	0.92	0.0080	3.45	0.125	0.0180	0.0024	0.98	0.0040	**1.32**
PosterLlama† [133]	✗	–	–	–	–	–	–	0.154	0.0135	0.0008	0.97	0.0030	2.21
LayoutPrompter† [134]	✓	0.220	0.0169	0.0006	0.91	0.0003	3.42	0.251	0.0179	**0.0004**	0.89	**0.0002**	4.59
VASCAR (GPT-4o)	✓	0.129	**0.0091**	**0.0002**	**0.99**	**0.0002**	**3.14**	0.141	**0.0102**	0.0005	**0.99**	**0.0002**	5.69
VASCAR (Gemini)	✓	**0.113**	0.0117	0.0013	0.98	0.0003	3.34	0.125	0.0122	0.0010	0.98	0.0007	6.27

4.4.2 Input-unconstrained Generation

We followed the experimental setup of RALF [3] to compare ours with others over the input-unconstrained generation task in Table 4.1. We observed that on PKU, VASCAR achieved the best performance across almost all metrics, significantly outperforming even fine-tuned base-

[3]FID is excluded because it is for a distribution but not a sample.

Table 4.2: Unconstrained generation results on the unannotated test split. Best and second-best results are **bold** and with underline.

	PKU unannotated				CGL unannotated			
Method	Occ ↓	Rea ↓	Und ↑	Ove ↓	Occ ↓	Rea ↓	Und ↑	Ove ↓
CGL-GAN [127]	0.191	0.0312	0.32	0.0690	0.481	0.0568	0.26	0.2690
DS-GAN [124]	0.180	0.0301	0.52	0.0260	0.435	0.0563	0.29	0.0710
ICVT [174]	0.189	0.0317	0.48	0.2920	0.446	0.0425	0.67	0.3010
LayoutDM [152]	0.165	0.0285	0.38	0.2010	0.421	0.0506	0.49	0.0690
Autoreg [3]	0.154	0.0274	0.35	0.0220	0.384	0.0427	0.76	0.0580
RALF [3]	0.133	0.0231	0.87	0.0180	0.336	0.0397	0.93	0.0270
VASCAR (Ours)	**0.132**	**0.0151**	**0.98**	**0.0002**	**0.289**	**0.0300**	**0.98**	**0.0007**

lines. In CGL, VASCAR also achieved competitive performance on Rea, Ove, and Und. In terms of FID, we found a large degradation, which we attribute to the fact that the model requires different numbers of self-corrections for easy and difficult samples. For simple samples, the VLM can generate layouts that meet the metrics within the initial few iterations. However, as the iterations increase, the model begins to overly adjust to the suggestions from the suggester, leading to results that increasingly deviate from the true layout, thus causing the FID to worsen. It is known that FID tends to perform better with slight noise added to the layout [175], and we believe that this deterioration in FID does not necessarily explain the quality of the generation.

Table 4.2 shows the performance comparison on the unannotated test split of PKU and CGL. VASCAR outperformed all other models across all metrics. While all compared models experienced performance degradation on these unannotated splits compared to the annotated ones due to the domain gap between inpainted and clean canvases, VASCAR demonstrated remarkable robustness to this domain gap, particularly in terms of the Und and Ove metrics.

Qualitative Comparison. Figure 4.3 shows the visual comparison of several baselines including Autoreg, RALF, and LayoutPrompter vs. our VASCAR across different self-correction

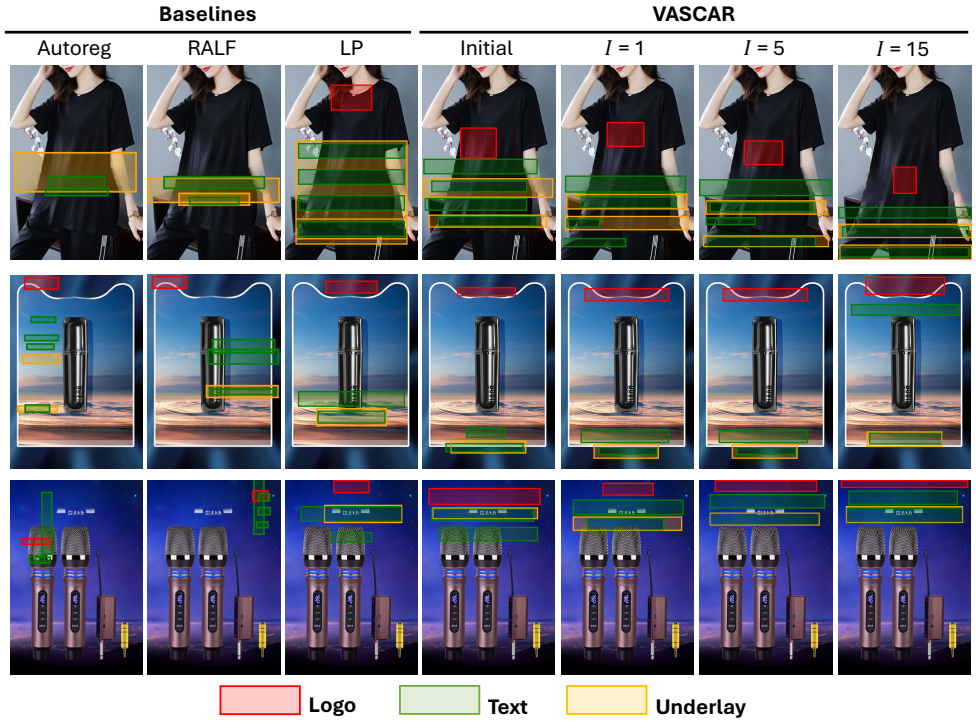


Figure 4.3: Visual comparison of baselines and VASCAR with different values of I.

I including Initial i.e. $I = 0$. We confirm that VASCAR can generate well-arranged layouts avoiding overlapping and occluding the main content. VASCAR's results improve progressively with iterations.

Trends of each metric. As shown in Figure 4.4, all metrics improve as the number of self-correction iterations increases, except for FID.

Human evaluation. To further demonstrate the effectiveness of VASCAR, we conducted a user study on the PKU test split. comparing it against strong baselines: RALF [3], Layout-Prompter [134], and Initial. We invited 15 participants to evaluate 30 groups, selecting one preferred choice from each group. The user study results, based on voting percentages, are shown in Figure 4.5. VASCAR outperforms other baselines significantly according to human evaluation.

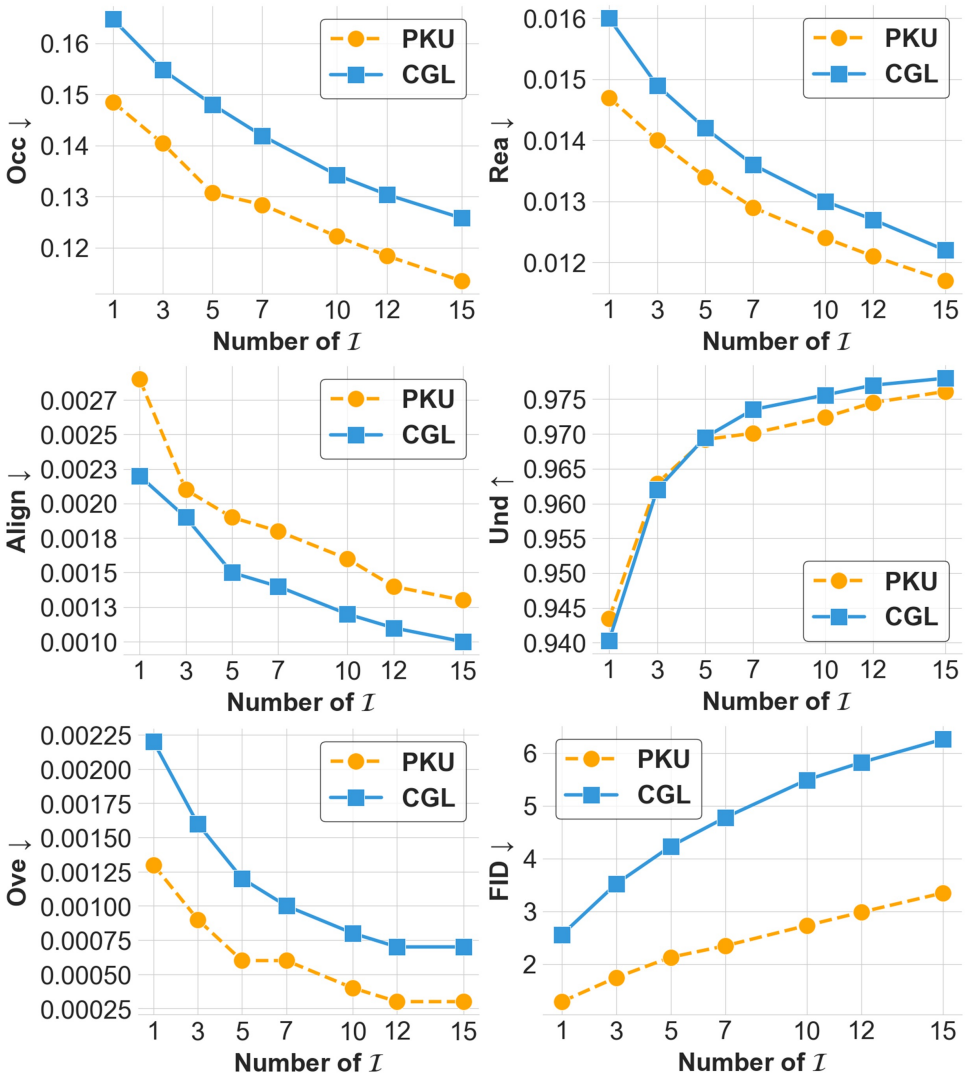


Figure 4.4: The trends of each metric based on the number of self-correction I.

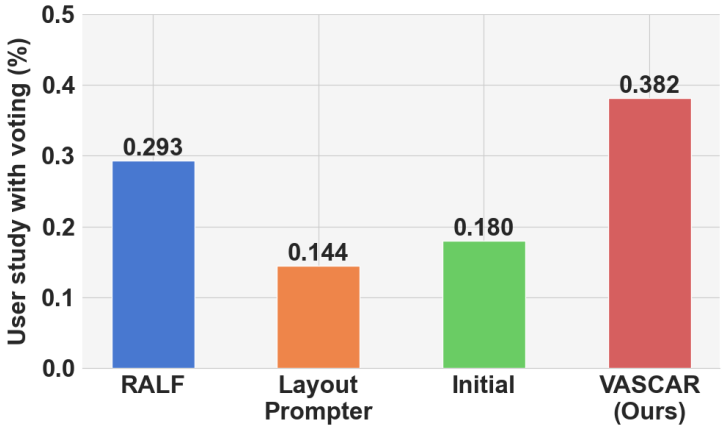


Figure 4.5: User study results on PKU test split.

4.4.3 Input-constrained Generation

We evaluated VASCAR on the five input-constrained generation tasks followed by Horita et al. [3]:

Category → Size + Position (C → S + P) task specifies the category and the number of visual elements to place.

Category + Size → Position (C + S→P) specifies the size of elements to generate in addition to the C → S + P task.

Completion refers to a task where the model is given all information about partial element (including category, size, and position) and asked to complete the layout.

Refinement represents the task of improving a perturbed layout, which is generated by adding Gaussian noise of mean 0 and variance 1 to the ground-truth layout [176].

Relationship specifies the positional and size relationships between elements.

We adopted the text prompt preprocessing inspired by [134], allowing VASCAR to handle these tasks suitable for content-aware layout generation. The details of the prompt can be found in Appendix.

As shown in Table 4.3, we found that VASCAR performs well across several input-constrained generation tasks. On PKU, VASCAR even outperforms the ground-truth layout (Real Data in the table) on certain metrics, proving its effectiveness. We observed that VAS-

Table 4.3: Quantitative result of five constrained generation tasks on the PKU and CGL test splits. Best and second-best results are in **bold** and underline, respectively.

	PKU					CGL				
	Content		Graphic			Content		Graphic		
Method	Occ ↓	Rea ↓	Und ↑	Ove ↓	FID↓	Occ ↓	Rea ↓	Und ↑	Ove ↓	FID↓
Real Data	0.112	0.0102	0.99	0.0009	1.58	0.125	0.0170	0.98	0.0002	0.79
C → S + P										
CGL-GAN	0.132	0.0158	0.48	0.0380	11.47	0.140	0.0213	0.65	0.0470	23.93
LayoutDM	0.152	0.0201	0.46	0.1720	20.56	0.127	0.0192	0.79	0.0260	3.39
Autoreg	0.135	0.0167	0.43	0.0280	10.48	0.124	0.0188	0.89	0.0150	1.36
RALF	0.124	0.0138	0.90	0.0100	**2.21**	0.126	0.0180	0.97	0.0060	**0.50**
VASCAR (GPT-4o)	0.117	**0.0094**	**1.00**	**0.0002**	3.01	0.139	**0.0099**	**0.99**	**0.0002**	4.86
VASCAR (Gemini)	**0.107**	0.0100	0.99	0.0003	2.82	**0.123**	0.0111	0.98	0.0005	5.51
C + S → P										
CGL-GAN	0.129	0.0155	0.48	0.0430	9.11	0.129	0.0202	0.75	0.0270	6.96
LayoutDM	0.143	0.0185	0.45	0.1220	24.90	0.127	0.0190	0.82	0.0210	2.18
Autoreg	0.137	0.0169	0.46	0.0280	5.46	0.127	0.0191	0.88	0.0130	0.47
RALF	0.125	0.0138	0.87	0.0100	**0.62**	0.128	0.0185	**0.96**	0.0060	**0.21**
VASCAR (GPT-4o)	0.123	0.0117	**0.90**	**0.0009**	1.11	0.140	0.0132	0.88	**0.0003**	1.67
VASCAR (Gemini)	**0.104**	**0.0107**	0.88	0.0018	2.21	**0.122**	**0.0123**	0.85	0.0028	2.39
Completion										
CGL-GAN	0.150	0.0174	0.43	0.0610	25.67	0.174	0.0231	0.21	0.1820	78.44
LayoutDM	0.135	0.0175	0.35	0.1340	21.70	0.127	0.0192	0.76	0.0200	3.19
Autoreg	0.125	0.0161	0.42	0.0230	5.96	**0.124**	0.0185	0.91	0.0110	2.33
RALF	0.120	0.0140	0.88	0.0120	**1.58**	0.126	0.0185	0.96	0.0050	**1.04**
VASCAR (GPT-4o)	0.120	**0.0063**	**1.00**	**0.0004**	6.19	0.137	**0.0068**	**0.99**	**0.0005**	5.57
VASCAR (Gemini)	**0.119**	0.0097	0.99	0.0006	4.74	0.135	0.0098	0.98	0.0009	5.18
Refinement										
CGL-GAN	0.122	0.0141	0.39	0.0900	6.40	0.124	0.0182	0.86	0.0240	1.20
LayoutDM	0.115	0.0174	0.40	0.0630	2.86	0.127	0.0196	0.75	0.0150	1.98
Autoreg	0.131	0.0171	0.40	0.0220	5.89	0.126	0.0183	0.89	0.0040	0.15
RALF	0.113	0.0109	0.95	0.0040	**0.13**	0.126	0.0176	**0.98**	0.0020	**0.14**
VASCAR (GPT-4o)	0.108	**0.0072**	**0.99**	**0.0005**	1.18	0.125	**0.0091**	0.95	**0.0008**	0.59
VASCAR (Gemini)	**0.104**	0.0095	0.97	0.0010	0.32	**0.114**	0.0096	0.96	0.0013	0.37
Relationship										
Autoreg	0.140	0.0177	0.44	0.0280	10.61	0.127	0.0189	0.88	0.0150	1.28
RALF	0.122	0.0141	0.85	0.0090	2.23	**0.126**	0.0184	**0.95**	0.0060	**0.55**
VASCAR (GPT-4o)	0.151	0.0139	0.92	0.0011	**1.94**	0.153	0.0139	0.91	**0.0004**	2.61
VASCAR (Gemini)	**0.119**	**0.0117**	**0.96**	**0.0008**	2.00	0.132	**0.0123**	**0.95**	0.0011	3.72

CAR's FID scores are lower than those of RALF on some tasks, which we consider is related to the weight λ_m in the layout scorer; excessive emphasis on the Occ and Rea metrics increases the distance between the generated layout and the ground-truth layout. This is further evidenced by its significantly better performance on content-related metrics compared to the ground-truth layouts.

4.4.4 Ablation Study

This section studies the impact of our design choices on PKU test split. Due to API limitations, I is set to 5.

The impact of multi-modal input. We argue that the visual content determines the performance of layout generation model. Therefore, we compared using only text input and different types of image input (the same image type was also used in self-correction). Table 4.4 shows that using only text input shows a significant gap in content metrics compared to multi-modal input. Regarding different image types, rendered images achieved the best performance in content metrics and comparable performance in graphic metrics, demonstrating that rendered images enhance VLMs' ability to generate layouts effectively.

Table 4.4: Multi-modal analysis for VASCAR on the PKU test split.

	Content		Graphic			
Setting	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Rendered Image (Ours)	**0.1304**	**0.0134**	**0.0017**	0.97	0.0012	2.13
Text-only	0.1529	0.0153	0.0024	0.97	0.0009	1.91
Saliency Map	0.1356	0.0140	0.0023	0.97	0.0009	**1.69**
Inpainting Image	0.1312	0.0137	0.0022	**0.98**	**0.0004**	2.37
Original Poster	0.1315	**0.0134**	0.0023	**0.98**	0.0006	2.34

Impact of the number of ICL examples. The number of ICL examples plays a crucial role in the performance of VLMs. When $M = 0$ (zero-shot), the models often fail to produce

reasonable results [134]. To explore this effect, we evaluate VASCAR using different numbers of ICL examples. The results are summarized in Table 4.5. Specifically, we experiment with $M = 1$, 3, 5, and 10. Our findings indicate that as M increases, performance improves across almost all metrics, with the exception of FID and Ove.

Table 4.5: Comparison of results across various experimental settings on the PKU test split. "Initial" means without self-correction. "w/o" means without. The blue and red values indicate improved and degraded metrics compared to VASCAR.

	Content		Graphic			
Setting	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Number of ICL Examples (M)						
1	0.2014	0.0184	0.0026	0.84	0.0017	**1.45**
3	0.1617	0.0160	0.0020	0.90	0.0016	2.27
5	0.1466	0.0150	0.0018	0.95	**0.0006**	2.25
10 (Ours)	**0.1304**	**0.0134**	**0.0017**	**0.97**	0.0012	2.13
Number of Output Candidates ($\|Y_q\|$)						
1	0.1659	0.0180	0.0023	0.89	0.0011	2.29
3	0.1393	0.0150	**0.0017**	0.96	0.0010	2.24
5 (Ours)	0.1304	0.0134	**0.0017**	**0.97**	0.0012	2.13
10	**0.1225**	**0.0127**	0.0025	**0.97**	**0.0008**	**2.04**
Ablation Study on λ						
VASCAR (Ours)	0.1304	0.0134	**0.0017**	**0.97**	0.0012	2.13
Initial	0.2080	0.0218	0.0040	0.97	0.0013	1.05
w/o Occ	0.1709	**0.0114**	0.0022	**0.97**	**0.0002**	**1.95**
w/o Rea	0.1284	0.0162	0.0024	**0.97**	0.0008	2.03
w/o Ove	0.1301	0.0137	0.0018	**0.97**	0.0030	2.23
w/o Und	**0.1237**	0.0130	0.0023	0.68	0.0007	2.14

Impact of the number of output candidates. We examine the effect of the number of output candidates ($|Y_q|$) in Table 4.5. Our findings indicate that as $|Y_q|$ increases, VASCAR's perfor-

mance improves accordingly. Considering the trade-off between performance and efficiency, we set $|Y_q| = 5$ as the default setting for VASCAR.

Impact of λ on the layout scorer. We evaluate the impact of the hyperparameter λ_m on the layout scorer by the following configurations: (1) *Without Occ*, (2) *Without Rea*, (3) *Without Ove*, and (4) *Without Und*. The results are summarized in Table 4.5. Our findings reveal that omitting a specific λ_m for the layout scorer leads to a significant degradation in the corresponding metric. While other metrics may show slight improvements in some cases, they remain optimized compared to the "Initial." Therefore, directly optimizing specific metrics is a reasonable approach.

In-depth analysis. Figure 4.6 shows an example with multiple candidate layout (i.e., Rank 1 to Rank 7). The colored frame of each poster indicates identical or similar layouts grouped by clustering [177] based on IoU among layouts; the same-colored frames mean that they are in the same cluster. In earlier steps, the layouts show diversity and converge into similar or near-duplicate layouts after some iterations.

Table 4.6: Comparison of post-hoc constraint optimization with LACE [4] and our VASCAR on the PKU test split. Blue and red numbers show improved and degraded metrics compared to Initial.

	Content		Graphic			
Setting	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Initial	0.210	0.0218	0.0040	0.97	0.0013	1.05
Initial + LACE	0.209	0.0233	0.0034	0.44	0.0002	1.06
VASCAR (Ours)	0.130	0.0134	0.0017	0.97	0.0012	2.13

VASCAR vs. constraint optimization. We further question whether correction similar to VASCAR is achievable via constraint-optimization techniques [4, 142] that explicitly minimize the layout metrics, such as Align. We compare VASCAR and the post-processing introduced in LACE [4] applied to our initial layouts (i.e., by the initial prompt). LACE considers Align and

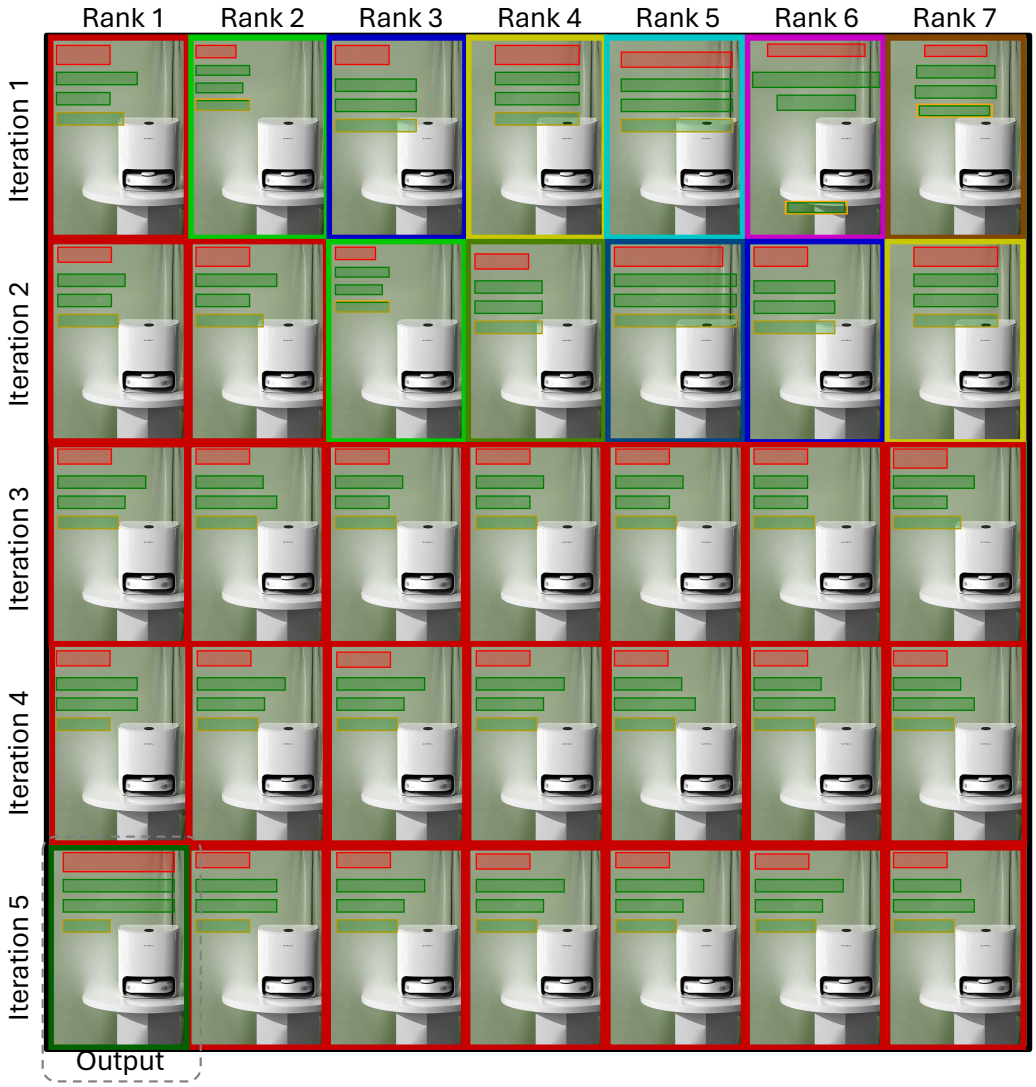


Figure 4.6: Visualization of the candidate pools under self-correction process with VASCAR. Best viewed in color.

Ove, which are differentiable, as the loss and minimize them with gradient decent. Obviously, as seen in Table 4.6, LACE's post-processing failed in improving the non-differentiable metrics. VASCAR can be seen as a gradient-free variant of such post-processing, and it has a clear advantage of reducing non-differential metrics, avoiding severe degradation thanks to VLM's tendency to generate plausible layouts.

Table 4.7: Comparison of results under different settings for ICL example selection on the PKU test split.

	Content		Graphic			
Setting	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
IoU (Ours)	**0.1304**	**0.0134**	**0.0017**	**0.97**	0.0012	**2.13**
Random	0.1740	0.0178	0.0018	**0.97**	**0.0002**	10.72
DreamSim [178]	0.1603	0.0149	0.0018	0.96	0.0010	3.77

Impact of ICL Example Retriever We evaluated different approaches to ICL retrieval. *Random* means selecting ICL examples randomly from $\mathcal{D}$. *DreamSim* is to use DreamSim [178] retriever in align with Horita et al. [3]. *IoU* refers to use IoU of saliency map. The results are shown in Table 4.7, indicating that *Random* selection achieved the worst performance, demonstrating that the choice of ICL examples is critical and should not be overlooked. Interestingly, retrieving ICL examples with the human-like *DreamSim* approach also led to inferior performance compared to the *IoU* selection. We argue that, in a training-free framework, selecting ICL examples based on IoU can implicitly enforce spatial information and compensate for the lack of *content constraint* within the textual prompt.

Impact of temperature on VLM. Temperature controls the level of randomness in token selection during generation. Lower temperatures are suited for prompts requiring deterministic or structured responses, while higher temperatures encourage more diverse and creative outputs [7]. To investigate its impact on VASCAR, we experimented with different temperature settings, with results presented in Table 4.8. Our findings show that when the temperature is

Table 4.8: Comparison of results across different temperature settings of VLM on the PKU test split.

	Content		Graphic			
Temperature	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
0	0.1956	0.0210	**0.0005**	0.76	0.0032	2.52
0.7	0.1482	0.0162	0.0010	**0.97**	0.0010	**2.00**
1.4 (Ours)	**0.1304**	0.0134	0.0017	**0.97**	0.0012	2.13
2.0	0.1319	**0.0131**	0.0027	0.96	**0.0009**	2.12

set to 0, the VLM tends to generate fixed outputs, lacking diversity and resulting in significant degradation across almost all metrics except for Align. At a temperature of 0.7, the results are also suboptimal. Conversely, increasing the temperature to an upper bound of 2.0 leads to a substantial decline in Occ and Align metrics. These observations highlight that selecting an appropriate temperature is crucial for achieving effective and creative layout generation with VASCAR.

Distribution of generated layouts. We analyse the layouts generated by VASCAR ($I = 15$) for the unconstrained task on the PKU and CGL test splits. The coordinates of the generated layouts, including left, top, width, and height, are restored to the original image size (513×750), as shown in Figure 4.7. The results indicate that the generated layouts are predominantly positioned in the upper-left region (0–100 pixels) with a large width (typically 400–500 pixels). From the figures showing the distribution of the top coordinates, there is sparse distribution in the middle regions of the canvas, suggesting that VASCAR tends to avoid placing layouts in these areas, which are often occupied by the main content.

4.4.5 Prompt Examples

For various constrained layout generation tasks, we adopt different prompts as outlined in [134]. Here, we provide detailed examples of these prompts (excluding images for simplicity) for the

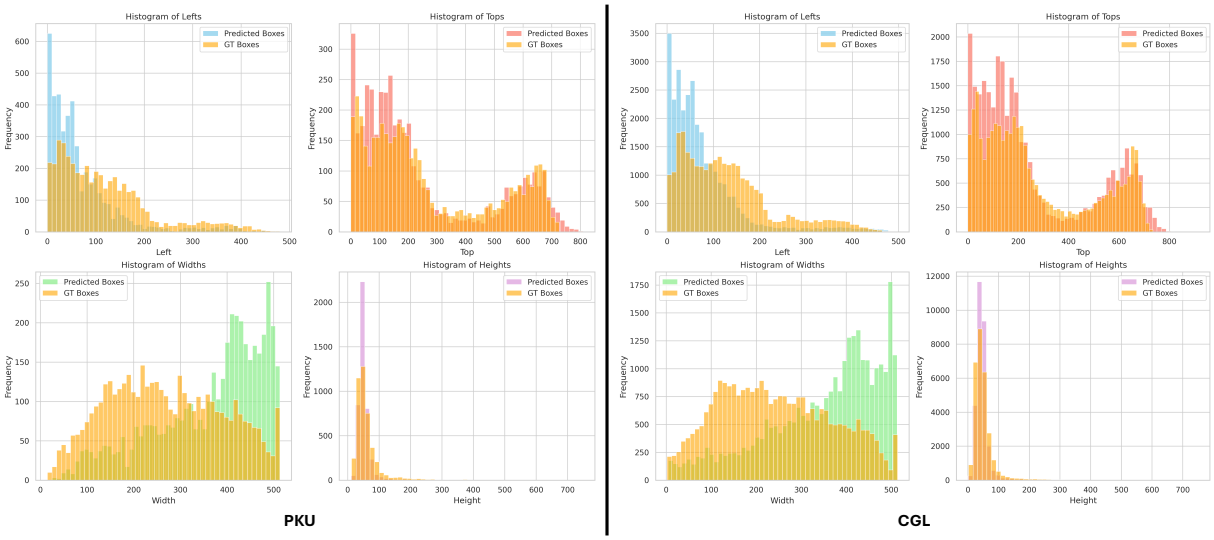


Figure 4.7: The distribution of generated layout results (Predicted Boxes) (coordinates: left, top, width, and height) and ground truth (GT Boxes) for the unconstrained layout generation task on the PKU and CGL test splits.

following tasks: **C** $\rightarrow$ **S + P** (Table 4.8), **C + S** $\rightarrow$ **P** (Table 4.9), **Completion** (Table 4.10), **Refinement** (Table 4.11), and **Relationship** (Table 4.12). Additionally, we present an optimized prompt example in Table 4.13.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout generation
Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 19px, top 57px, width 59px, height 63px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 7px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 38px; top: 23px; width: 26px; height: 9px"></div>
<div class="text" style="index: 2; left: 13px; top: 34px; width: 74px; height: 7px"></div>
<div class="text" style="index: 3; left: 16px; top: 41px; width: 63px; height: 7px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 24px, top 59px, width 52px, height 68px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | text 4 | underlay 5 | underlay 6

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 3px; top: 2px; width: 30px; height: 7px"></div>
<div class="text" style="index: 1; left: 25px; top: 15px; width: 53px; height: 5px"></div>
<div class="text" style="index: 2; left: 13px; top: 24px; width: 76px; height: 11px"></div>
<div class="text" style="index: 3; left: 36px; top: 42px; width: 24px; height: 4px"></div>
<div class="text" style="index: 4; left: 74px; top: 62px; width: 11px; height: 13px"></div>
<div class="underlay" style="index: 5; left: 31px; top: 40px; width: 40px; height: 9px"></div>
<div class="underlay" style="index: 6; left: 69px; top: 57px; width: 20px; height: 22px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5

Figure 4.8: A prompt example of **C** → **S + P** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout generation conditioned on given element types and sizes
Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 20px, top 57px, width 55px, height 71px
Element Type and Size Constraint: logo 0 29 6 | text 1 84 11 | text 2 51 6 | text 3 18 6 | text 4 18 6 | underlay 5 84 7 | underlay 6 29 29

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 9px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 8px; top: 23px; width: 84px; height: 11px"></div>
<div class="text" style="index: 2; left: 24px; top: 37px; width: 51px; height: 6px"></div>
<div class="text" style="index: 3; left: 16px; top: 59px; width: 18px; height: 6px"></div>
<div class="text" style="index: 4; left: 15px; top: 66px; width: 18px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 7px; top: 36px; width: 84px; height: 7px"></div>
<div class="underlay" style="index: 6; left: 10px; top: 52px; width: 29px; height: 29px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 23px, top 62px, width 57px, height 60px | left 26px, top 65px, width 18px, height 20px
Element Type and Size Constraint: logo 0 62 12 | text 1 86 14 | text 2 38 11 | underlay 3 62 16

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 18px; top: 3px; width: 62px; height: 12px"></div>
<div class="text" style="index: 1; left: 7px; top: 19px; width: 86px; height: 14px"></div>
<div class="text" style="index: 2; left: 31px; top: 35px; width: 38px; height: 11px"></div>
<div class="underlay" style="index: 3; left: 19px; top: 33px; width: 62px; height: 16px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type and Size Constraint: logo 0 27 9 | text 1 58 14 | text 2 47 5 | text 3 21 6 | underlay 4 53 9 | underlay 5 43 10

Figure 4.9: A prompt example of **C + S→P** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout completion

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 26px, top 59px, width 47px, height 61px
Partial Layout: logo 0 11 7 80 10

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 11px; top: 7px; width: 80px; height: 10px"></div>
<div class="text" style="index: 1; left: 7px; top: 23px; width: 86px; height: 18px"></div>
<div class="text" style="index: 2; left: 18px; top: 47px; width: 65px; height: 9px"></div>
<div class="text" style="index: 3; left: 35px; top: 130px; width: 29px; height: 6px"></div>
<div class="underlay" style="index: 4; left: 13px; top: 45px; width: 75px; height: 12px"></div>
<div class="underlay" style="index: 5; left: 27px; top: 126px; width: 47px; height: 13px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 20px, top 63px, width 56px, height 60px
Partial Layout: logo 0 26 18 48 9

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 26px; top: 18px; width: 48px; height: 9px"></div>
<div class="text" style="index: 1; left: 10px; top: 30px; width: 81px; height: 12px"></div>
<div class="text" style="index: 2; left: 24px; top: 45px; width: 52px; height: 5px"></div>
<div class="text" style="index: 3; left: 77px; top: 64px; width: 13px; height: 13px"></div>
<div class="underlay" style="index: 4; left: 19px; top: 43px; width: 62px; height: 9px"></div>
<div class="underlay" style="index: 5; left: 71px; top: 58px; width: 24px; height: 24px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Partial Layout: logo 0 37 10 27 9

Figure 4.10: A prompt example of **Completion** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout refinement

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 20px, top 63px, width 56px, height 60px
Noise Layout: logo 0 27 22 46 7 | text 1 9 30 81 13 | text 2 22 46 52 5 | text 3 78 62 13 15 | underlay 4 20 42 61 9 | underlay 5 73 58 24 27

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 26px; top: 18px; width: 48px; height: 9px"></div>
<div class="text" style="index: 1; left: 10px; top: 30px; width: 81px; height: 12px"></div>
<div class="text" style="index: 2; left: 24px; top: 45px; width: 52px; height: 5px"></div>
<div class="text" style="index: 3; left: 77px; top: 64px; width: 13px; height: 13px"></div>
<div class="underlay" style="index: 4; left: 19px; top: 43px; width: 62px; height: 9px"></div>
<div class="underlay" style="index: 5; left: 71px; top: 58px; width: 24px; height: 24px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 20px, top 57px, width 55px, height 71px
Noise Layout: logo 0 38 10 30 3 | text 1 8 22 84 10 | text 2 24 38 51 7 | text 3 15 58 18 5 | text 4 17 68 18 6 | underlay 5 9 36 83 7 | underlay 6 10 52 29 28

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 9px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 8px; top: 23px; width: 84px; height: 11px"></div>
<div class="text" style="index: 2; left: 24px; top: 37px; width: 51px; height: 6px"></div>
<div class="text" style="index: 3; left: 16px; top: 59px; width: 18px; height: 6px"></div>
<div class="text" style="index: 4; left: 15px; top: 66px; width: 18px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 7px; top: 36px; width: 84px; height: 7px"></div>
<div class="underlay" style="index: 6; left: 10px; top: 52px; width: 29px; height: 29px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Noise Layout: logo 0 37 6 26 10 | text 1 23 27 58 14 | text 2 25 49 48 7 | text 3 46 130 20 6 | underlay 4 24 45 55 9 | underlay 5 28 127 43 12

Figure 4.11: A prompt example of **Refinement** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout conditioned on given element relationships
'A left B' means that the center coordinate of A is to the left of the center coordinate of B.
'A right B' means that the center coordinate of A is to the right of the center coordinate of B.
'A top B' means that the center coordinate of A is above the center coordinate of B.
'A bottom B' means that the center coordinate of A is below the center coordinate of B.
'A center B' means that the center coordinate of A and the center coordinate of B are very close.
'A smaller B' means that the area of A is smaller than the area of B.
'A larger B' means that the area of A is larger than the area of B.
'A equal B' means that the area of A and the area of B are very close.
Here, center coordinate = (left + width / 2, top + height / 2), area = width * height

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 26px, top 59px, width 47px, height 61px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5
Element Relationship Constraint: underlay 5 smaller logo 0 | text 3 bottom text 2

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>\n<div
class="logo" style="index: 0; left: 11px; top: 7px; width: 80px; height: 10px"></div>\n<div
class="text" style="index: 1; left: 7px; top: 23px; width: 86px; height: 18px"></div>\n<div
class="text" style="index: 2; left: 18px; top: 47px; width: 65px; height: 9px"></div>\n<div
class="text" style="index: 3; left: 35px; top: 130px; width: 29px; height: 6px"></div>\n<div
class="underlay" style="index: 4; left: 13px; top: 45px; width: 75px; height: 12px"></div>\n<div
class="underlay" style="index: 5; left: 27px; top: 126px; width: 47px; height: 13px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 25px, top 60px, width 51px, height 64px
Element Type Constraint: text 0 | text 1 | text 2 | text 3 | text 4 | underlay 5 | underlay 6 | underlay 7
Element Relationship Constraint: text 1 top canvas | underlay 6 center canvas | text 4 smaller text 3 | underlay 6 center text 3 | underlay 7 larger text 4 | underlay 7 center text 4 | underlay 7 right underlay 5

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="text" style="index: 0; left: 2px; top: 24px; width: 96px; height: 15px"></div>
<div class="text" style="index: 1; left: 11px; top: 41px; width: 77px; height: 5px"></div>
<div class="text" style="index: 2; left: 6px; top: 49px; width: 25px; height: 7px"></div>
<div class="text" style="index: 3; left: 39px; top: 49px; width: 25px; height: 7px"></div>
<div class="text" style="index: 4; left: 71px; top: 50px; width: 24px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 4px; top: 48px; width: 29px; height: 8px"></div>
<div class="underlay" style="index: 6; left: 36px; top: 48px; width: 29px; height: 8px"></div>
<div class="underlay" style="index: 7; left: 68px; top: 48px; width: 29px; height: 8px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5
Element Relationship Constraint: text 3 bottom canvas | underlay 5 bottom canvas | text 1 larger logo 0 | text 3 smaller logo 0

Figure 4.12: A prompt example of **Relationship** task.

(Initial Prompt)
Here are the generated results of Query Image: Red is Logo; Green is Text; Orange is Underlay.

Output Image 1:
<html>
<body>

</body>
</html>



Output Image 1

The score is:
0.8929
Please refine the layout following the instruction below:
- Ensure that the underlay is properly covered by text or logos, enhancing visual appeal.
- Reduce the overlap between content constraint areas and layout elements to avoid occluding important content.
- Enhance the readability of text elements by adjusting font size, spacing, or positioning.

Output Image 2:
<html>
<body>

</body>
</html>



Output Image 2

The score is:
0.8433
Please refine the layout following the instruction below:
- Ensure that the underlay is properly covered by text or logos, enhancing visual appeal.
- Reduce the overlap between content constraint areas and layout elements to avoid occluding important content.

……

Please propose new layout; Score is high as much as possible.
(Generation by LVLM…)

Figure 4.13: A prompt example of optimized prompt.

4.5 Summary

This chapter introduces VASCAR, a novel *training-free* framework for content-aware layout generation that utilizes VLMs through a visual-aware self-correction mechanism like a designer. The method iteratively refines layouts by incorporating feedback from both rendered visual contents and automatic metrics, improving layout quality without additional training. Extensive experiments and user study demonstrate that VASCAR excels in generating high-quality layouts and adapts well to different datasets and different VLMs, offering a versatile solution for content-aware layout generation tasks.

Limitations. In exchange for improved layout quality, VASCAR needs more API calls and token consumption than other single-shot API-based methods, resulting in consumption of more money and time for generation. However, the consumption is affordable at the moment; VASCAR costs less than 100 USD and 1.5 hours for 1,000-sample generation for the PKU experiments.

Chapter 5

Conclusion

In this thesis, we investigated data- and parameter-efficient adaptation methods for foundation models and their effective applications. We demonstrate our methodologies across three progressive scenarios ranging from parameter tuning to training-free interaction. First, we showed that PEFT is effective for challenging 3D medical diagnosis by adapting 2D VFMs via anatomy-guided slices from a parameter-tuning perspective. Second, we applied parameter-efficient visual prompting from an input-level perspective to steer frozen foundation models within the VICL paradigm. Finally, we explored data- and parameter-efficient adaptation methods for proprietary generative models, using the challenging task of content-aware layout generation as a case study. We demonstrate that VICL serves as a data-efficient paradigm for task definition, while visual-aware self-correction provides an effective optimization strategy from a training-free perspective.

In Chapter 2, we demonstrated the efficacy of PEFT for the challenging diagnosis of PF-ILD. By devising *ILD-Slider* with slice-level adapters, we showed that utilizing anatomy-guided slices is data-efficient, significantly reducing the need for high-cost annotation and computational resources typical of 3D medical data. Furthermore, we established that 2D VFMs can effectively capture volumetric dependencies with minimal trainable parameters, outperforming fully fine-tuned baselines.

In Chapter 3, we shifted our focus to input-level optimization for foundation models through

the VICL paradigm. With *E-InMeMo*, we proved that optimizing learnable prompting to bridge visual discrepancies, while keeping model parameters frozen, significantly enhances performance in both natural and medical domains. This method offers an efficient adaptation approach for adapting foundation models to downstream tasks without requiring weight updates to the backbone model.

Finally, in Chapter 4, we addressed the challenge of applying data- and parameter-efficient methods to proprietary generative models, taking content-aware layout generation as a representative task. We proposed *VASCAR*, a training-free visual self-correction framework that effectively steers frozen generative models through iterative visual feedback and heuristic guidelines, without accessing gradients or internal architectures. We demonstrated that VICL serves as a data-efficient method for task definition, and that training-free visual self-correction acts as an effective optimization strategy to unlock the latent generative capabilities of proprietary foundation models.

Beyond presenting three methods, we emphasize that they address a common practical challenge in real deployments: the degree of access to a foundation model varies across applications. In some settings, the model is locally deployable and limited task-specific training is allowed under privacy or governance constraints; in others, the model is provided as a service and users can only control inputs and prompts; in the most restrictive case, the model is accessible only through a black-box API with no gradients or parameter updates. Viewed through this lens, the three approaches form a coherent spectrum of data- and parameter-efficient adaptation.

Specifically, when lightweight task-specific training is feasible, PEFT provides an efficient way to specialize a model with a small trainable parameter budget (Chapter 2). When weight updates are not desirable or not permitted and adaptation must be achieved through the user interface, pixel-level prompt enhancement improves the robustness of VICL under imperfect demonstrations (Chapter 3). In the most restrictive setting, where only black-box access is available, training-free steering via iterative automatic feedback enables continual refinement without any tuning (Chapter 4).

While this thesis has established a robust framework for efficient adaptation through ILD-

Slider, E-InMeMo, and VASCAR, the landscape of foundation models is rapidly evolving toward dynamic, collaborative, and agentic systems. Current PEFT methods primarily rely on static parameter allocation. To overcome this rigidity, future work should explore architectures utilizing Mixture-of-Experts (MoE) strategies [179] to dynamically allocate adapter parameters based on input complexity. Furthermore, given the emergence of foundation models capable of unified multi-modal input and output processing [7, 180, 181], developing data- and parameter-efficient methodologies tailored for these unified next-token prediction architectures remains a critical research frontier.

Acknowledgements

I would like to express my deepest gratitude to Professor Yuta Nakashima from SANKEN, the University of Osaka, Professor Hajime Nagahara from the D3 Center, the University of Osaka, Associate Professor Hong Liu from Xiamen University, and Assistant Professor Bowen Wang from SANKEN, the University of Osaka, for their unwavering support, invaluable guidance, and continuous encouragement throughout my doctoral studies.

I am also especially thankful to Dr. Ryota Yoshihashi, Dr. Shunsuke Kitada, and Mr. Atsuki Osanai for their mentorship during my internship and research collaboration at LY Corporation. I deeply appreciate their willingness to share their expertise, provide thoughtful criticism, and patiently guide me through challenging projects. Their support has played a crucial role in both my personal and professional growth.

The medical data utilized in this thesis was collected from the University of Osaka Hospital and the National Hospital Organization Osaka Toneyama Medical Center. I would like to acknowledge the support and instruction provided by Professor Toshihiro Takeda, Assistant Professor Shoya Wada, Assistant Professor Kento Sugimoto, Dr. Takayuki Niitsu, Assistant Professor Kiyoharu Fukushima, Dr. Hiroshi Kida, and Professor Yasushi Matsumura.

I am profoundly grateful for the financial assistance provided by the Fellowship for Integration of Knowledge with Society (「社会と知の統合」を実現するイノベーション博士人材フェローシップ). This fellowship enabled me to focus entirely on my research without the burden of financial concerns, and I truly appreciate their belief in my potential and investment in my academic development.

Finally, I would like to thank my family and friends for their unwavering support, under-

standing, and companionship during the often strenuous journey of the PhD program. Their constant encouragement, patience, and care have been a tremendous source of strength and comfort, and I am deeply appreciative of all they have done to help me reach this milestone.

Reference

[1] Tao Li, Chin-Yi Cheng, Amber Xie, Gang Li, and Yang Li. Revision matters: Generative design guided by revision edits. *arXiv preprint arXiv:2406.18559*, 2024.

[2] Peitong Duan, Jeremy Warner, Yang Li, and Bjoern Hartmann. Generating automatic feedback on ui mockups with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2024.

[3] Daichi Horita, Naoto Inoue, Kotaro Kikuchi, Kota Yamaguchi, and Kiyoharu Aizawa. Retrieval-augmented layout transformer for content-aware layout generation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 67–76, 2024.

[4] Jian Chen, Ruiyi Zhang, Yufan Zhou, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *Proceedings of the International Conference on Learning Representations*, 2024.

[5] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[6] GPT-4o. `https://openai.com/index/hello-gpt-4o/`. Last accessed 21 February, 2025.

[7] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[10] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. In *Advances in Neural Information Processing Systems*, Vol. 35, pp. 25005–25017, 2022.

[11] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Proceedings of the European Conference on Computer Vision*, pp. 493–510, 2022.

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the International Conference on Computer Vision*, pp. 4015–4026, 2023.

[13] Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General object foundation model for images and videos at scale. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 3783–3795, 2024.

[14] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the International Conference on Machine Learning*, pp. 2790–2799, 2019.

[16] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, Vol. 55, No. 9, pp. 1–35, 2023.

[17] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 487–503, 2021.

[18] Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *Proceedings of the International Conference on Learning Representations*, 2024.

[19] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

[20] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 2225–2240, 2022.

[21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision*, pp. 709–727, 2022.

[22] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *Proceedings of the International Conference on Machine Learning*, pp. 9614–9624, 2020.

[23] Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 24224–24235, 2023.

[24] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems*, Vol. 35, pp. 16664–16678, 2022.

[25] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, Vol. 130, No. 9, pp. 2337–2348, 2022.

[26] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022.

[27] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, Vol. 132, No. 2, pp. 581–595, 2024.

[28] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, et al. Sam-med3d: A vision foundation model for general-purpose segmentation on volumetric medical images. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 36, No. 10, pp. 17599–17612, 2025.

[29] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023.

[30] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901, 2020.

[31] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 17773–17794, 2023.

[32] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2655–2671, 2022.

[33] Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. In *Proceedings of the*

Conference on Empirical Methods in Natural Language Processing, pp. 10136–10148, 2023.

[34] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1423–1436, 2023.

[35] Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 819–862, 2022.

[36] Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1935–1952, 2023.

[37] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 13484–13508, 2023.

[38] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 5316–5330, 2022.

[39] Jiahao Zhang, Ryota Yoshihashi, Shunsuke Kitada, Atsuki Osanai, and Yuta Nakashima. Vascar: Content-aware layout generation via visual-aware self-correction. *arXiv preprint arXiv:2412.04237*, 2024.

[40] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 6830–6839, 2023.

[41] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *Proceedings of the International Conference on Computer Vision*, pp. 1130–1140, 2023.

[42] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 22861–22872, 2024.

[43] Jiahao Zhang, Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Instruct me more! random prompting for visual in-context learning. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 2597–2606, 2024.

[44] AU Wells and N Hirani. Interstitial lung disease guideline. *Thorax*, Vol. 63, No. Suppl 5, pp. v1–v58, 2008.

[45] Marlies Wijsenbeek, Michael Kreuter, Amy Olson, Aryeh Fischer, Elisabeth Bendstrup, Christopher D Wells, Christopher P Denton, Baher Mounir, Leila Zouad-Lejour, Manuel Quaresma, et al. Progressive fibrosing interstitial lung diseases: current practice in diagnosis and management. *Current Medical Research and Opinion*, Vol. 35, No. 11, pp. 2015–2024, 2019.

[46] Xianqiu Chen, Jian Guo, Dong Yu, Bing Jie, and Ying Zhou. Predictors of mortality in progressive fibrosing interstitial lung diseases. *Frontiers in Pharmacology*, Vol. 12, p. 754851, 2021.

[47] Mouhamad Nasser, Sophie Larrieu, Loïc Boussel, Salim Si-Mohamed, Fabienne Diaz, Sébastien Marque, Jacques Massol, Didier Revel, Francoise Thivolet-Bejui, Lara Cha-

labreysse, Delphine Maucort-Boulch, and Vincent Cottin. Epidemiology and mortality of non-idiopathic pulmonary fibrosis (ipf) progressive fibrosing interstitial lung disease (pf-ild) using the french national health insurance system (snds) database in france: the progress study. *European Respiratory Journal*, Vol. 56, No. suppl 64, 2020.

[48] Gerald J Roth, Rudolf Binder, Florian Colbatzky, Claudia Dallinger, Rozsa Schlenker-Herceg, Frank Hilberg, Stefan-Lutz Wollin, and Rolf Kaiser. Nintedanib: from discovery to the clinic. *Journal of Medicinal Chemistry*, Vol. 58, No. 3, pp. 1053–1063, 2015.

[49] De-yu Li, Xin Liu, Jing-yi Huang, Wen-lu Hang, Gu-ran Yu, and Yong Xu. Impact of antifibrotic therapy on disease progression, all-cause mortality, and risk of acute exacerbation in non-ipf fibrosing interstitial lung diseases: evidence from a meta-analysis of randomized controlled trials and prospective controlled studies. *Therapeutic Advances in Respiratory Disease*, Vol. 18, p. 17534666241232561, 2024.

[50] Norah Asiri, Muhammad Hussain, Fadwa Al Adel, and Nazih Alzaidi. Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial Intelligence in Medicine*, Vol. 99, p. 101701, 2019.

[51] Bowen Wang, Toshihiro Takeda, Kento Sugimoto, Jiahao Zhang, Shoya Wada, Shozo Konishi, Shirou Manabe, Katsuki Okada, and Yasushi Matsumura. Automatic creation of annotations for chest radiographs based on the positional information extracted from radiographic image reports. *Computer Methods and Programs in Biomedicine*, Vol. 209, p. 106331, 2021.

[52] Bowen Wang, Jiuyang Chang, Yiming Qian, Guoxin Chen, Junhao Chen, Zhouqiang Jiang, Jiahao Zhang, Yuta Nakashima, and Hajime Nagahara. Direct: Diagnostic reasoning for clinical notes via large language models. In *Advances in Neural Information Processing Systems*, Vol. 37, pp. 74999–75011, 2024.

[53] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, Vol. 37, No. 3, pp. 362–386, 2020.

[54] Jiahao Zhang and Anqi Hu. Analyzing green view index and green view index best path using google street view and deep learning. *Journal of Computational Design and Engineering*, Vol. 9, No. 5, pp. 2010–2023, 2022.

[55] Sweta Bhattacharya, Siva Rama Krishnan Somayaji, Thippa Reddy Gadekallu, Mamoun Alazab, and Praveen Kumar Reddy Maddikunta. A review on deep learning for future smart cities. *Internet Technology Letters*, Vol. 5, No. 1, p. e187, 2022.

[56] Heang-Ping Chan, Lubomir M Hadjiiski, and Ravi K Samala. Computer-aided diagnosis in the era of deep learning. *Medical Physics*, Vol. 47, No. 5, pp. e218–e227, 2020.

[57] Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical Image Analysis*, Vol. 102, p. 103547, 2025.

[58] Zhongyi Shui, Jianpeng Zhang, Weiwei Cao, Sinuo Wang, Ruizhe Guo, Le Lu, Lin Yang, Xianghua Ye, Tingbo Liang, Qi Zhang, and Ling Zhang. Large-scale and fine-grained vision-language pre-training for enhanced CT image understanding. In *Proceedings of the International Conference on Learning Representations*, 2025.

[59] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *Preceedings of the International Conference on Learning Representations*, 2022.

[60] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapt-

ing transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 46–54, October 2020.

[61] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4582–4597, 2021.

[62] Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 5203–5212, 2021.

[63] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4222–4235, 2020.

[64] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 61–68, 2022.

[65] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *Advances in Neural Information Processing Systems*, Vol. 35, pp. 26462–26477, 2022.

[66] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 19434–19445, 2023.

[67] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

[68] Jiahao Zhang, Bowen Wang, Hong Liu, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. E-inmemo: Enhanced prompting for visual in-context learning. *Journal of Imaging*, Vol. 11, No. 7, p. 232, 2025.

[69] Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, Vol. 8, No. 11, p. 713, 2020.

[70] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[71] Qing Li, Weidong Cai, and David Dagan Feng. Lung image patch classification with automatic feature learning. In *Preceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6079–6082, 2013.

[72] Taro Sakamoto, Tomoi Furukawa, Hoa HN Pham, Kishio Kuroda, Kazuhiro Tabata, Yukio Kashima, Ethan N Okoshi, Shimpei Morimoto, Andrey Bychkov, and Junya Fukuoka. A collaborative workflow between pathologists and deep learning for the evaluation of tumour cellularity in lung adenocarcinoma. *Histopathology*, Vol. 81, No. 6, pp. 758–769, 2022.

[73] Emerson Nithiyaraj and Arivazhagan Selvaraj. Ctsc-net: an effectual ct slice classification network to categorize organ and non-organ slices from a 3-d ct image. *Neural Computing and Applications*, Vol. 34, No. 24, pp. 22141–22156, 2022.

[74] Shaohua Li, Yong Liu, Xiuchao Sui, Cheng Chen, Gabriel Tjio, Daniel Shu Wei Ting, and Rick Siow Mong Goh. Multi-instance multi-scale cnn for medical image classification. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 531–539, 2019.

[75] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, Vol. 35, pp. 23716–23736, 2022.

[76] Jiahao Zhang, Bowen Wang, Hong Liu, Yuta Nakashima, and Hajime Nagahara. Panicl: Mitigating over-reliance on single prompt in visual in-context learning. *arXiv preprint arXiv:2509.21926*, 2025.

[77] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2021.

[78] Taiki Furukawa, Shintaro Oyama, Hideo Yokota, Yasuhiro Kondoh, Kensuke Kataoka, Takeshi Johkoh, Junya Fukuoka, Naozumi Hashimoto, Koji Sakamoto, Yoshimune Shiratori, et al. A comprehensible machine learning tool to differentially diagnose idiopathic pulmonary fibrosis from other chronic interstitial lung diseases. *Respirology*, Vol. 27, No. 9, pp. 739–746, 2022.

[79] Christopher J Ryerson, Eric Vittinghoff, Brett Ley, Joyce S Lee, Joshua J Mooney, Kirk D Jones, Brett M Elicker, Paul J Wolters, Laura L Koth, Talmadge E King Jr, et al. Predicting survival across chronic interstitial lung disease: the ild-gap model. *Chest*, Vol. 145, No. 4, pp. 723–728, 2014.

[80] Simon LF Walsh, John A Mackintosh, Lucio Calandriello, Mario Silva, Nicola Sverzellati, Anna Rita Larici, Stephen M Humphries, David A Lynch, Helen E Jo, Ian Glaspole, et al. Deep learning–based outcome prediction in progressive fibrotic lung disease using high-resolution computed tomography. *American Journal of Respiratory and Critical Care Medicine*, Vol. 206, No. 7, pp. 883–891, 2022.

[81] Takayuki Niitsu, Kiyoharu Fukushima, Sho Komukai, So Takata, Yuko Abe, Takuro Nii, Tomoki Kuge, Shinichi Iwakoshi, Takayuki Shiroyama, Kotaro Miyake, et al. Real-world impact of antifibrotics on prognosis in patients with progressive fibrosing interstitial lung disease. *RMD Open*, Vol. 9, No. 1, p. e002667, 2023.

[82] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pp. 6105–6114, 2019.

[83] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[85] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[86] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[87] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the International Conference on Computer Vision*, pp. 618–626, 2017.

[88] David A Lynch, Nicola Sverzellati, William D Travis, Kevin K Brown, Thomas V Colby, Jeffrey R Galvin, Jonathan G Goldin, David M Hansell, Yoshikazu Inoue, Takeshi Johkoh, et al. Diagnostic criteria for idiopathic pulmonary fibrosis: a fleischner

society white paper. *The Lancet Respiratory Medicine*, Vol. 6, No. 2, pp. 138–153, 2018.

[89] Joseph Jacob, Brian J Bartholmai, Srinivasan Rajagopalan, Maria Kokosi, Arjun Nair, Ronald Karwoski, Simon LF Walsh, Athol U Wells, and David M Hansell. Mortality prediction in idiopathic pulmonary fibrosis: evaluation of computer-based ct analysis with conventional severity measures. *European Respiratory Journal*, Vol. 49, No. 1, 2017.

[90] Simon LF Walsh, Anand Devaraj, Juan Ignacio Enghelmayer, Kazuma Kishi, Rafael S Silva, Nina Patel, Milton D Rossman, Claudia Valenzuela, and Carlo Vancheri. Role of imaging in progressive-fibrosing interstitial lung diseases. *European Respiratory Review*, Vol. 27, No. 150, 2018.

[91] Yanpeng Sun, Qiang Chen, Jian Wang, Jingdong Wang, and Zechao Li. Exploring effective factors for improving visual in-context learning. *IEEE Transactions on Image Processing*, 2025.

[92] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 19133–19143, 2023.

[93] Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *Proceedings of the International Conference on Learning Representations*, 2019.

[94] Paarth Neekhara, Shehzeen Hussain, Jinglong Du, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Cross-modal adversarial reprogramming. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 2427–2435, 2022.

[95] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A sur-

vey on in-context learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128, 2024.

[96] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out*, pp. 100–114, 2022.

[97] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8086–8098, 2022.

[98] Patrick H Winston. Learning and reasoning by analogy. *Communications of the ACM*, Vol. 23, No. 12, pp. 689–703, 1980.

[99] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *Proceedings of the International Conference on Machine Learning*, pp. 20841–20855, 2022.

[100] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, Vol. 24, No. 240, pp. 1–113, 2023.

[101] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2791–2809, 2022.

[102] Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Workshop*, 2022.

[103] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, Vol. 35, pp. 24824–24837, 2022.

[104] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Preceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5687–5711, 2023.

[105] Shengnan An, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Jian-Guang Lou, and Dongmei Zhang. How do in-context examples affect compositional generalization? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 11027–11052, 2023.

[106] Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. On the compositional generalization gap of in-context learning. In *Proceedings of the BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 272–280, 2022.

[107] Hsi-Ai Tsao, Lei Hsiung, Pin-Yu Chen, Sijia Liu, and Tsung-Yi Ho. AutoVP: An automated visual prompting framework and benchmark. In *Proceedings of the International Conference on Learning Representations*, 2024.

[108] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning with rules for text classification. *AI Open*, Vol. 3, pp. 182–192, 2022.

[109] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

[110] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

[111] Zhen Liu Irfan Essa Amirreza Shaban, Shray Bansal and Byron Boots. One-shot learning for semantic segmentation. In *Proceedings of the British Machine Vision Conference*, pp. 167.1–167.13, 2017.

[112] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, Vol. 88, pp. 303–338, 2010.

[113] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *Proceedings of the International Conference on MultiMedia Modeling*, pp. 451–462, 2019.

[114] Guanghui Yue, Houlu Xiao, Hai Xie, Tianwei Zhou, Wei Zhou, Weiqing Yan, Baoquan Zhao, Tianfu Wang, and Qiuping Jiang. Dual-constraint coarse-to-fine network for camouflaged object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 34, No. 5, pp. 3286–3298, 2023.

[115] David Gutman, Noel CF Codella, Emre Celebi, Brian Helba, Michael Marchetti, Nabin Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1605.01397*, 2016.

[116] Jiahui Zhong, Wenhong Tian, Yuanlun Xie, Zhijia Liu, Jie Ou, Taoran Tian, and Lei Zhang. Pmfsnet: Polarized multi-scale feature self-attention network for lightweight medical image segmentation. *Computer Methods and Programs in Biomedicine*, p. 108611, 2025.

[117] Wei Suo, Lanqing Lai, Mengyang Sun, Hanwang Zhang, Peng Wang, and Yanning Zhang. Rethinking and improving visual prompt selection for in-context learning segmentation. In *Proceedings of the European Conference on Computer Vision*, pp. 18–35, 2024.

[118] Chengming Xu, Chen Liu, Yikai Wang, Yuan Yao, and Yanwei Fu. Towards global optimal visual in-context learning prompt selection. In *Advances in Neural Information Processing Systems*, Vol. 37, pp. 74945–74965, 2024.

[119] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *Proceedings of the International Conference on Learning Representations Workshop*, 2018.

[120] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pp. 740–755, 2014.

[121] Simon Lok and Steven Feiner. A survey of automated layout techniques for information presentations. *Proceedings of SmartGraphics*, Vol. 2001, pp. 61–68, 2001.

[122] Yong Shi, Mengyu Shang, and Zhiquan Qi. Intelligent layout generation based on deep generative models: A comprehensive survey. *Information Fusion*, p. 101940, 2023.

[123] Shunan Guo, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. Vinci: An Intelligent Graphic Design System for Generating Advertising Posters. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2021.

[124] Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout. In

Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 6018–6026, 2023.

[125] Ali Jahanian, Jerry Liu, Qian Lin, Daniel Tretter, Eamonn O'Brien-Strain, Seungyon Claire Lee, Nic Lyons, and Jan Allebach. Recommendation system for automatic design of magazine covers. In *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 95–106, 2013.

[126] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. Automatic generation of visual-textual presentation layout. *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 12, No. 2, pp. 1–22, 2016.

[127] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pp. 4995–5001, 2022.

[128] Fengheng Li, An Liu, Wei Feng, Honghe Zhu, Yaoyu Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Junjie Shen, Zhangang Lin, et al. Relation-aware diffusion model for controllable poster layout generation. In *Proceedings of the International Conference on Information and Knowledge Management*, pp. 1249–1258, 2023.

[129] Shang Chai, Liansheng Zhuang, Fengying Yan, and Zihan Zhou. Two-stage content-aware layout generation for poster designs. In *Proceedings of the International Conference on Multimedia*, pp. 8415–8423, 2023.

[130] Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. Spot the error: Non-autoregressive graphic layout generation with wireframe locator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, pp. 3413–3421, 2024.

[131] Shoma Iwai, Atsuki Osanai, Shunsuke Kitada, and Shinichiro Omachi. Layout-corrector: Alleviating layout sticking phenomenon in discrete diffusion model. In *Proceedings of the European Conference on Computer Vision*, pp. 92–110, 2024.

[132] Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. LayoutNUWA: Revealing the hidden layout expertise of large language models. In *Proceedings of the International Conference on Learning Representations*, 2024.

[133] Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of language model to content-aware layout generation. In *Proceedings of the European Conference on Computer Vision*, pp. 451–468, 2024.

[134] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutprompter: Awaken the design ability of large language models. In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 43852–43879, 2024.

[135] Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v for automatic image design and generation. In *Proceedings of the European Conference on Computer Vision*, pp. 167–184, 2024.

[136] Daichi Haraguchi, Naoto Inoue, Wataru Shimoda, Hayato Mitani, Seiichi Uchida, and Kota Yamaguchi. Can gpts evaluate graphic design based on design principles? In *Proceedings of the SIGGRAPH Asia Technical Communications*, 2024.

[137] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 373–380, 1992.

[138] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 249–256, 1990.

[139] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023.

[140] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[141] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *Proceedings of the International Conference on Learning Representations*, 2024.

[142] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *Proceedings of the International Conference on Multimedia*, pp. 88–96, 2021.

[143] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-pagegraphic designs. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 20, No. 8, pp. 1200–1213, 2014.

[144] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Designscape: Design with interactive layout suggestions. In *Proceedings of the Annual Conference on Human Factors in Computing Systems*, pp. 1221–1224, 2015.

[145] Jianan Li, Tingfa Xu, Jianming Zhang, Aaron Hertzmann, and Jimei Yang. LayoutGAN: Generating graphic layouts with wireframe discriminator. In *Proceedings of the International Conference on Learning Representations*, 2019.

[146] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout gan for automatic graphic design. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 27, No. 10, pp. 4039–4048, 2020.

[147] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the International Conference on Computer Vision*, pp. 9895–9904, 2019.

[148] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the International Conference on Computer Vision*, pp. 5481–5489, 2021.

[149] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[150] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the International Conference on Computer Vision*, pp. 1004–1014, 2021.

[151] Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutFormer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the International Conference on Computer Vision*, pp. 18403–18412, 2023.

[152] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 10167–10176, 2023.

[153] Shang Chai, Liansheng Zhuang, and Fengying Yan. Layoutdm: Transformer-based diffusion model for layout generation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 18349–18358, 2023.

[154] Julian Jorge Andrade Guerreiro, Naoto Inoue, Kento Masui, Mayu Otani, and Hideki Nakayama. Layoutflow: Flow matching for layout generation. In *Proceedings of the European Conference on Computer Vision*, pp. 56–72, 2024.

[155] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics*, Vol. 38, No. 4, pp. 1–15, 2019.

[156] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[157] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[158] Weixi Feng, Wanrong Zhu, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Xuehai He, S Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 18225–18250, 2023.

[159] Mengyao Chen, Xinghua Zhang, Junhao Zhang, Quangang Li, and Tingwen Liu. Empowering llms for multi-page layout generation via consistency-oriented in-context learning. In *Proceedings of the International Conference on Information and Knowledge Management*, pp. 3679–3683, 2024.

[160] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *Proceedings of the International Conference on Learning Representations*, 2024.

[161] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A prompt pattern

catalog to enhance prompt engineering with chatgpt. In *Proceedings of the Conference on Pattern Languages of Programs*, 2023.

[162] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *Proceedings of the International Conference on Learning Representations*, 2023.

[163] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 16477–16508, 2023.

[164] Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. Lever: Learning to verify language-to-code generation with execution. In *Proceedings of the International Conference on Machine Learning*, pp. 26106–26128, 2023.

[165] Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 6327–6336, 2024.

[166] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 46534–46594, 2023.

[167] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *Proceedings of the International Conference on Learning Representations*, 2024.

[168] Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, Vol. 12, pp. 1417–1440, 2024.

[169] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Preceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 840–854, 2022.

[170] Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 5168–5186, 2022.

[171] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 7479–7489, 2019.

[172] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *Proceedings of the European Conference on Computer Vision*, pp. 38–56, 2022.

[173] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022.

[174] Yunning Cao, Ye Ma, Min Zhou, Chuanbin Liu, Hongtao Xie, Tiezheng Ge, and Yuning Jiang. Geometry aligned variational transformer for image-conditioned layout generation. In *Proceedings of the International Conference on Multimedia*, pp. 1561–1571, 2022.

[175] Mayu Otani, Naoto Inoue, Kotaro Kikuchi, and Riku Togashi. Ltsim: Layout transportation-based similarity measure for evaluating layout generation. *arXiv preprint arXiv:2407.12356*, 2024.

[176] Soliha Rahman, Vinoth Pandian Sermuga Pandian, and Matthias Jarke. Ruite: Refining ui layout aesthetics using transformer encoder. In *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 81–83, 2021.

[177] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, Vol. 96, pp. 226–231, 1996.

[178] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 50742–50768, 2023.

[179] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[180] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[181] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

List of Publications

Journal Publications (related to this thesis)

1. **Jiahao Zhang**, Bowen Wang, Hong Liu, Liangzhi Li, Yuta Nakashima, Hajime Nagahara. "E-InMeMo: Enhanced Prompting for Visual In-Context Learning." *Journal of Imaging, vol. 11, no. 7 (2025): 232.*

2. **Jiahao Zhang**, Shoya Wada, Kento Sugimoto, Takayuki Niitsu, Kiyoharu Fukushima, Hiroshi Kida, Bowen Wang, Shozo Konishi, Katsuki Okada, Yuta Nakashima, Toshihiro Takeda. "ILD-Slider: A Parameter-Efficient Model for Identifying Progressive Fibrosing Interstitial Lung Disease from Chest CT Slices." *Journal of Imaging, vol. 11, no. 10 (2025): 353.*

International Conference (related to this thesis)

1. **Jiahao Zhang**, Bowen Wang, Liangzhi Li, Yuta Nakashima, Hajime Nagahara. "Instruct Me More! Random Prompting for Visual In-Context Learning." In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV), pp. 2597-2606, 2024.*

2. **Jiahao Zhang**, Ryota Yoshihashi, Shunsuke Kitada, Atsuki Osanai, Yuta Nakashima. "VASCAR: Content-Aware Layout Generation via Visual-Aware Self-Correction." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2026 (Submitted).*

3. **Jiahao Zhang**, Bowen Wang, Hong Liu, Yuta Nakashima, Hajime Nagahara. "PANICL: Mitigating Over-Reliance on Single Prompt in Visual In-Context Learning." In *Proceedings of the International Conference on Learning Representations, 2026 (Submitted).*

Domestic Conference (related to this thesis)

1. **張 家豪**, 和田 聖哉, 杉本 賢人, 武田 理宏, 岡田 佳築, 真鍋 史朗, 小西 正三, 松村 泰志, "二次元画像分類モデルによる間質性肺炎代表断面像抽出手法の開発," 第 41 回医療情報学連合大会, 2021.

2. **張 家豪**, 和田 聖哉, 杉本 賢人, 岡田 佳築, 真鍋 史朗, 小西 正三, 松村 泰志, 武田 理宏, "画像認識モデルによる胸部 CT の解剖学的断面像抽出手法を介した間質性肺炎の特徴断面像の同定." 第 4 回日本メディカル AI 学会学術集会, 2022.

3. **張 家豪**, 和田 聖哉, 杉本 賢人, 岡田 佳築, 真鍋 史朗, 小西 正三, 木田 博, 福島 清春, 服部 聡, 松村 泰志, 武田 理宏, "画像認識モデルによる胸部 CT の解剖学的断面像抽出手法を介した間質性肺炎予後予測モデルの開発." 第 42 回医療情報学連合大会, 2022.

4. **Jiahao Zhang**, Ryota Yoshihashi, Shunsuke Kitada, Atsuki Osanai, Yuta Nakashima, "VASCAR: Content-Aware Layout Generation via Visual-Aware Self-Correction." 第 28 回 画像の認識・理解シンポジウム (MIRU2025), 2025.

Journal Publications (not related to this thesis)

1. Bowen Wang, Toshihiro Takeda, Kento Sugimoto, **Jiahao Zhang**, Shoya Wada, Shozo Konishi, Shirou Manabe, Katsuki Okada, Yasushi Matsumura. "Automatic creation of annotations for chest radiographs based on the positional information extracted from radiographic image reports." *Computer Methods and Programs in Biomedicine, vol. 209 (2021): 106331.*

2. **Jiahao Zhang**, and Anqi Hu, "Analyzing green view index and green view index best path using Google street view and deep learning." *Journal of Computational Design and Engineering, vol. 9, no. 5 (2022): 2010.*

International Conference (not related to this thesis)

1. Bowen Wang, Toshihiro Takeda, Kento Sugimoto, **Jiahao Zhang**, Shoya Wada, Shozo Konishi, Shirou Manabe, Yasushi Matsumura. "Semi Supervised Learning of Nodule Detection in Chest Radiographs." In *The 11th Asia-Pacific Association for Medical Informatics, 2020.*

2. Bowen Wang, Jiuyang Chang, Yiming Qian, Guoxin Chen, Junhao Chen, Zhouqiang Jiang, **Jiahao Zhang**, Yuta Nakashima, Hajime Nagahara. "DiReCT: Diagnostic Reasoning for Clinical Notes via Large Language Models." In *Advances in Neural Information Processing Systems, vol. 37 (2024): 74999.*

Awards

1. The 11th Asia-Pacific Association for Medical Informatics 2020, Best Paper Award
2. 第 4 回日本メディカル AI 学会学術集会, 優秀ポスター演題賞
3. 第 28 回画像の認識・理解シンポジウム（MIRU2025）, 学生奨励賞