AdaptPrompt with Diffusion Set: A Unified Framework for Generalizable Deepfake Detection

by

Yichen Jiang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Yichen Jiang 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Deepfake detection focuses on identifying synthetic media. It has critical applications in cybersecurity, misinformation mitigation, digital forensics, and media authentication. Recent developments in deepfake detection have achieved impressive performance, leveraging deep learning models to distinguish between real and synthetic content. However, recent developments in diffusion-based generative models and publicly available tools such as Stable Diffusion and DALL-E pose immediate challenges to existing detection techniques. Diffusion models generate photorealistic and high-resolution content with less observable or detectable artifacts and are, therefore, more difficult to detect using traditional deepfake detection techniques.

In this thesis, we present a comprehensive study of existing deepfake detection techniques and adapting large vision-language models, i.e., CLIP, to generalizable deepfake detection. We introduce the Diffusion Set, a new dataset of 100k diffusion-generated fake images and 100k real images. Our experiments reveal that detectors trained on Diffusion Set outperform detectors trained on GAN-based datasets. To further enhance deepfake detection, we introduce a new transfer learning strategy that learns randomly initialized prompts and a lightweight adapter network while having CLIP frozen. Extensive experiments confirm its efficiency, and we also investigate the impact of dropping specific CLIP layers on detection accuracy.

Our study utilizes the Diffusion Set for training and evaluates models on 25 unseen test sets, covering images synthesized by GAN-based models, diffusion-based models, and commercially available tools. Beyond large-scale training, we assess model performance in few-shot settings, where models are trained with only a small fraction of the dataset (e.g., 320 real and 320 fake images), providing insights into their adaptability under data constraints.

Additionally, we extend our analysis beyond classification by exploring image attribution and training models in a few-shot setting to attribute images to specific generators such as BigGAN, StarGAN, and Stable Diffusion. Our findings showcase the robustness of CLIP-based models in deepfake detection and their ability to generalize across unseen generative techniques. We also investigated the feasibility of using the same transfer learning strategy for attribution, with experimental results that demonstrate its high effectiveness in closed set attribution.

Acknowledgements

I would like to thank all the people who made this thesis possible. Specifically, I would like to express my sincere gratitude to Prof. Fakhri Karray for his continuous support and guidance. I am also profoundly grateful to Mohammed Talha Alam and Dr. Sohail Ahmed Khan for their generous and selfless support. I sincerely thank my thesis committee members, Prof. Ramadan El-Shatshat and Prof. Andrew Morton, for their valuable feedback and thoughtful evaluation, which greatly enhance this work.

Dedication

This is dedicated to everyone I love, everyone who loves me, and the sunset over Waterloo — Waterloo's sunset is fine.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures ix

List of Tables x

1 Introduction 1

1.1 Preface 1

1.2 Motivations 1

1.3 Problems and Challenges 2

1.4 Proposed Solutions 2

1.5 Contributions 3

1.6 Organization of Thesis 4

2 Background and Literature Review 5

2.1 Deepfake Image Generation 5

2.1.1 Generative Adversarial Networks(GANs) . . . 5
2.1.2 Diffusion Models(DMs) . . . 6
2.2 Deepfake Detection Approaches . . . 6
2.3 Pre-trained Vision-Language Models . . . 7
2.4 Transfer Learning for Vision-Langugae Models . . . 8
2.5 Summary . . . 9

3 Proposed Methodology **10**
3.1 Problem Formulation . . . 10
3.2 Baseline . . . 12
3.3 AdaptPrompt Transfer Learning Strategy . . . 13
3.3.1 Overview . . . 13
3.3.2 Method Details . . . 13
3.4 Summary . . . 16

4 Experimental Results **17**
4.1 Datasets . . . 17
4.1.1 Training . . . 17
4.1.2 Testing . . . 18
4.2 Diffusion Set . . . 18
4.2.1 Experimental Setup . . . 18
4.2.2 Experimental Results . . . 20
4.3 AdaptPrompt Experiments . . . 22
4.3.1 Experimental Setup . . . 22
4.3.2 Experimental Results . . . 22
4.3.3 Ablation Studies . . . 25
4.4 Attribution . . . 26
4.4.1 Experimental Setup . . . 26

4.4.2 Experimental Results 26
4.5 Unsuccessful Attempts 27
4.5.1 Experiments on Combination with One-Shot Test-Time Training . . 27
4.5.2 Experiments on Combination with NoisePrint and TruFor 27
4.6 Summary 29

5 Conclusion **30**

References **32**

APPENDICES **38**

A Original Data **39**

List of Figures

3.1 This figure showcases four transfer learning strategies for deepfake detection that were explored in the work of Khan et al.[35], including Linear Probing, Fine-tuning, Adapter Network and Prompt Tuning. At the bottom right, we annotated the number of trainable parameters. 11

3.2 This figure showcases our proposed transfer learning strategy, AdaptPrompt. At the bottom right, we annotated the number of trainable parameters. . . 11

4.1 Confusion matrix for the deepfake attribution task, where each entry represents the classification percentage for a given generator. The vertical axis (rows) corresponds to actual generator labels, and the horizontal axis (columns) corresponds to the predicted labels. 28

List of Tables

4.1 This table presents the statistics of the test datasets. For datasets that do not include real images, we use real images from the LAION dataset as a substitute. . . . 19

4.2 This table presents the average precision and accuracy achieved by four transfer learning strategies trained using ProGAN and Diffusion Set. . . . 20

4.3 This table showcases the scores achieved by the methods trained using a smaller-sized Diffusion Set. Results are organized based on the total number of training images: 20k, 40k, 60k, and 80k. Each dataset consists of an equal proportion of real and fake images; e.g. for a 20k dataset, we have 10k real and 10k fake images. . . . 21

4.4 This table showcases the performance of five transfer learning strategies trained on the complete Diffusion Set (including 100k real images and 100k fake images). AdaptPrompt_v0 represents the original version of AdaptPrompt, AdaptPrompt_v1 represents AdaptPrompt without the projection layer, and AdaptPrompt_v2 represents AdaptPrompt without the last transformer block. . . . 23

4.5 This table shows the performance scores of AdaptPrompt trained on 20k, 40k, 60k, 80k dataset. . . . 24

4.6 This table presents the results from our few-shot (32-shot) experiment. . . . 24

4.7 This table shows the scores achieved by fixed prompt AdaptPrompt and learnable prompt AdaptPrompt. . . . 25

4.8 Performance comparison between prompt-only AdaptPrompt and full AdaptPrompt. . . . 26

A.1 Original data on average precision. . . . 40

A.2 Original data on accuracy. . . . 40

Chapter 1

Introduction

1.1 Preface

Generative models have advanced rapidly in recent years, enabling the creation of highly realistic synthetic media. With the rise of deep generative techniques, particularly Generative Adversarial Networks (GANs) and diffusion models, it has become increasingly challenging to distinguish between real and artificially generated images and videos. While these models facilitated unprecedented progress in creative applications, content creation and enrichment of information, they also gave rise to severe ethical, security, and social problems. Deepfake has caused widespread misinformation, identity theft, and privacy violations, and this has given rise to the need for deepfake detection methods. With synthetic media getting more advanced by the day, traditional forensic methods are no longer capable of keeping pace with it. This has given rise to a requirement for advanced, generalizable, and scalable deepfake detection methods.

1.2 Motivations

Deepfake technology has introduced new possibilities for modern media, allowing anyone to generate their own artistic creations with just one click. Some of these works exhibit strong stylistic characteristics, while others are so realistic that they are nearly indistinguishable from real images. However, this technology also comes with inherent risks. Many fake images generated have reached a level where they are difficult to distinguish with the

naked eye, potentially leading to misinformation, privacy violations, and various ethical and security concerns[4, 9].

As deepfake techniques improve, their detection becomes more challenging. Traditional detection methods have been effective against GAN-generated deepfakes due to their ability to capture characteristic artifacts. However, they often struggle to generalize to diffusion-generated fakes and commercial tools, which produce more photorealistic results with fewer detectable artifacts[7, 34, 56]. To address this issue, researchers are exploring new approaches that leverage vision-language models(VLMs) to enhance detection accuracy and robustness[42]. Existing methods for transfer learning in VLMs focus on unimodal features, either textual features only or visual features only[35]. However, existing approaches do not fully leverage the multimodal capabilities of VLMs for deepfake detection.

This thesis investigates the potential of transfer learning in deepfake detection, particularly through integration of VLMs. We aim to develop a more effective and generalizable deepfake detection framework using both image and textual features.

1.3 Problems and Challenges

Our target in this thesis is to explore the use of transfer learning to leverage pre-trained vision language models (VLMs) for deepfake detection. The main challenge is that previous approaches have not fully utilized both the language and visual components of VLMs simultaneously, often utilizing only one of them at most.

1.4 Proposed Solutions

In this paper, we present a comprehensive analysis of traditional methods for deepfake detection and the adaptation of large vision-language models, specifically Contrastive Language-Image Pre-training(CLIP), for generalizable deepfake detection. We proposed a new dataset, Diffusion Set, which contains 100k diffusion-based models generated fake images and 100k real images. The experiments show that detectors trained by the Diffusion Set perform better than those trained by the GAN-based training set. We then proposed our new transfer learning strategy, which learns the randomly initialized prompts and an extra lightweight bottleneck layer (adapter network) and keeps the CLIP model frozen. We also performed extensive experiments to verify its effectiveness and further analyzed, through experiments, how removing specific layers of CLIP affects performance.

Our study employs a Diffusion Set for training and evaluates models on 25 unseen test sets comprising images synthesized using GAN-based models, diffusion-based models, and commercially available tools.

In addition to large-scale training, we investigate few-shot learning scenarios, where models are trained with a limited subset of the dataset (e.g., 320 real and 320 fake images). This allows us to assess the model adaptability of different strategies under data-scarce conditions. Furthermore, we extend our analysis beyond classification by training models in a few-shot setting to attribute images to specific generators, such as BigGAN, StarGAN, Stable Diffusion, etc. Our findings provide key insights into the robustness of CLIP-based models for deepfake detection and their ability to generalize across unseen generative techniques. We also explored the possibility of using the same transfer learning strategy for attribution. Experimental results show that this approach is highly effective for closed-set attribution.

1.5 Contributions

The main contributions of this thesis are summarized as follows:

- We perform a comprehensive analysis of the performance, generalizability, and computational efficiency of existing transfer learning strategies for deepfake detection tasks. Our evaluation is based on 25 distinct datasets that cover GAN-based generators, diffusion-based generators, and commercial tools.
- We propose AdaptPrompt, a novel transfer learning strategy for deepfake detection. By integrating vision-language models(VLMs), our method effectively utilizes both visual and contextual components of VLMs. Furthermore, we demonstrate that AdaptPrompt is not only effective for deepfake detection, but also holds potential for other transfer learning tasks, making it a versatile approach in broader applications.
- We introduce a new large-scale dataset, Diffusion Set, designed to train deepfake detectors. The dataset consists of 100k fake images generated by diffusion models and 100k real images sourced from the LSUN[55] dataset. We kept the object class distribution identical to real and fake images.
- Our proposed AdaptPrompt framework outperforms state-of-the-art deepfake detection methods in both average accuracy and average precision.

1.6 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 presents the background and review of the literature on the relevant deepfake detection approaches. In Chapter 3, we present the proposed methodology for the technical analysis of AdaptPrompt. Chapter 4 discusses the experimental setup and results to evaluate the proposed approaches, analysis points, and the corresponding discussion. Lastly, in Chapter 5, we present the summary of this thesis and discuss its related future research directions.

Chapter 2

Background and Literature Review

In this chapter, we will begin with the background of the generative models and the types of synthetic images in Section 2.1. Then, we will go through the previous methods of deepfake detection in Section 2.2. Finally, we will also introduce pre-trained vision-language models in Section 2.3 and transfer learning methods in Section 2.4.

2.1 Deepfake Image Generation

Synthetic images may involve modification on real images, for instance, utilizing Adobe Photoshop[3], or replacing a specific region like human face [1, 2], known as face swap techniques. Other synthetic images are generated by generative models from scratch. With provided input like random noise[29, 32], object information[5], or natural language prompts[41, 46, 51], the models could generate all pixels of images. The two mainstream models are generative adversarial networks(GANs) [22]and Diffusion Models(DMs)[52]. We will go through them respectively.

2.1.1 Generative Adversarial Networks(GANs)

GANs were first introduced by Goodfellow et al.[22] and have become a foundational technology for deepfake generation. GANs consist of two major parts: a generator that generates synthetic images and a discriminator that distinguishes synthesized images from natural images. During the training process, the performance of two parts will be improved jointly. Following the idea of GANs, more works[25, 30, 51] emerged to improve the training

process of GANs. For instance, conditional GANs[41, 54] allowed users to define generation parameters, increasing realism. BigGAN proposed by Brock et al.[5] serves as a class conditional image generator, which synthesizes images conditioned on specific class labels. ProGAN by Karras et al. [29] provides a stable training procedure which can produce high-quality images and increase image resolution over time. StyleGAN[32, 33, 31] makes convolutional kernels of the generators controllable by latent code, improving the quality and diversity of generated images. EG3D[6] can produce multiview-consistent renderings and detailed geometry of synthetic faces. Despite their success, GAN-generated images often exhibit detectable artificial fingerprints. As demonstrated in [14, 15], GAN-generated image artifacts are more easily spotted in the frequency domain and are clearly visible in the artificial fingerprint spectra, which makes model-specific detection plausible.

2.1.2 Diffusion Models(DMs)

Recent advancements have shifted towards Diffusion Models[52]. Diffusion models contain two interconnected stochastic processes. One process transforms natural images into random noise, and the other iteratively denoises the input noise into synthetic images. The diffusion model training process is more stable and generates images with less detectable artifacts. Similarly to the GAN family, there are also many variants of diffusion models. The slide introduced by Nichol et al.[40]uses a test encoder for conditional generation of descriptions in natural language based on vision-language models. Stable Diffusion [49], which is part of an open-source project, provides more possibilities for user-created content and derived products. While DMs are pushing the quality and diversity of synthetic images to the next level, they also bring more challenges to the deepfake detection task.

2.2 Deepfake Detection Approaches

With the rise and advancement of generative models, both commercial and open-source, there has been an increasing need to detect deepfake content. There is extensive literature on deepfake detection. Some methods target visible traces[17, 18, 38], such as asymmetries in human faces, excessive lighting, or perspective. Rossler et al.[50] focused on detecting manipulated facial images to target more precise needs. Some works[14, 15, 20] focus on the frequency domain and demonstrate that GAN-generated images are embedded with clear artificial fingerprints, which could be a tracing tool to discriminate between real and fake images. However, such classifiers exhibit limited generalizability. Studies[34, 58, 23] indicate this generalization dilemma.

Traditional detection methods often struggle to generalize across unseen generative models[7], particularly for diffusion-based images. Corvi et al. [11]found that models trained on GAN-generated images struggled with diffusion-generated images and vice versa. Ojha et al.[42] highlighted the challenge of detecting unseen deepfakes, noting that classifiers often treat non-GAN-generated images as real due to the absence of GAN-specific artifacts, i.e., real may be regarded as a sink label.

Wang[53] proposed an effective method to improve generalization for GANs. The authors trained a ResNet-50[26] with cross-entropy loss to perform binary classification to detect fake images. The model is trained with 360k ProGAN-generated[29] images as fake images and 360k LSUN[55] images as real images. The authors also included appropriate augmentation during training. This method significantly improves robustness and generalizes well on other variants of GAN models. Gragnaniello et al.[23] modified the structure of ResNet-50[26] to avoid downsampling in initial layers in order to better preserve artificial fingerprints. The authors also pointed out that the diversity of the training set improves generalization ability and suggested that deep-fake detectors should try to avoid information loss. The methods suggested by the authors include training the network on randomly cropped patches, using a fusion strategy to determine the final label for the entire image, and avoiding downsamplings in the first layers of the network.

Corvi et al. [11] trained the same model as in [23] with the dataset proposed by Wang et al.[53] and LDMs[49], separately. The authors suggested that while this classifier generalizes well when the test set is derived from the same generator family (GAN or DM), its performance drops when applied to different generator families. Ojha et al.[42] utilized pre-trained vision-language models (VLM) to improve generalization. With fixed CLIP[47] encoder, the authors trained a linear classifier on CLIP features using the training set as in [53, 23], achieving state-of-the-art(SOTA) results for both GAN and Diffusion model-generated images. We will discuss methods related to transfer learning strategies on pre-trained vision-language models with more details in Section 2.4

2.3 Pre-trained Vision-Language Models

Vision-language models (VLMs) learn a common representation space between textual and visual modalities. Typically, a VLM comprises an image encoder and a text encoder that embed images and text into a shared embedding space. During the training time, the model aligns image-text pairs by minimizing a contrastive loss such that matched pairs are close and unmatched pairs are far away. After training, VLMs, such as CLIP[47] and ALIGN[28], are able to accomplish a range of cross-modal tasks including image-to-text

retrieval, zero-shot classification and multi-modal reasoning by virtue of the established joint representations.

Contrastive Language-Image Pre-training (CLIP)[47] was trained on 400 million image-text pairs using contrastive learning to maximize the similarity between the images and the corresponding text captions. A Large-scale ImaGe and Noisy-text embedding(ALIGN)[28] used a billion noisy image-text pairs, demonstrating that dataset scale can compensate for noise. Both VLMs show a strong capability for diverse downstream tasks with zero-shot fine-tuning, while fine-tuning still brings extra benefits for specific tasks. LORA (Low-Rank Adaptation)[27] emerged as an efficient fine-tuning method, reducing computational complexity while maintaining model performance.

To achieve higher performance in specific downstream tasks, VLMs often require fine-tuning for downstream tasks like deepfake detection[36], compromising the robustness to distribution shifts[47]. Ensuring generalization across generative models remains an open challenge, particularly for diffusion-generated images.

2.4 Transfer Learning for Vision-Langugae Models

While directly fine-tuning large pre-trained vision-language models can be effective, it often suffers from limitations such as overfitting on limited datasets, high computational costs, and suboptimal generalization to unseen domains. To address these challenges, transfer learning strategies have emerged as a promising alternative, enabling efficient adaptation of these models to new tasks while leveraging their pre-trained knowledge. Transfer learning strategies aim to leverage knowledge from a source domain or task to improve the performance on a target domain or task, typically by reusing, adapting, or augmenting pre-trained models. Ojha et al. [42] directly extracted the fixed CLIP vision transformer (CLIP ViT) and applied linear probing and the nearest neighbor on it. Linear probing refers to training a shallow linear classifier on top of frozen features to evaluate their representational quality. With this method, the authors preserved the pre-trained knowledge of CLIP and adapted it specifically for the deepfake detection task. Zhou et al.[56] suggested that linear probing does not take advantage of the potential of the textual component of VLMs, and relying only on visual components may lead to suboptimal performance. Therefore, they introduced Context Optimization (CoOp), which enhances VLMs by injecting learnable vectors into prompts during fine-tuning. This approach, referred to as prompt tuning, adapts models for specific tasks without altering the underlying architecture. Prompt tuning improves flexibility and reduces computational overhead, making it suitable for deepfake detection tasks. Furthermore, the Adapter Network is introduced by

Gao et al.[21], which refers to a bottleneck layer to integrate new features while retaining the original CLIP features, enhancing transferability and preserving pre-trained knowledge during fine-tuning. Following these works, Khan et al. [35] compared four transfer learning strategies (fine-tuning the entire model, linear probing used in [42], Adapter Network introduced in [21], and Prompt Tuning proposed in [56]) to determine the most effective one, and the results demonstrated that Adapter Network and Prompt Tuning show superior performance and generalization compared to linear probing introduced in [42] and fine-tuning.

2.5 Summary

In this chapter, we reviewed the recent progress in deepfake detection. While traditional studies have made significant progress on the deepfake detection task on GAN-generated images, they often struggle to generalize across other generator families like diffusion-based and commercial tools. Regarding this problem, recent works have turned to transfer learning with vision-language models, leveraging the pre-trained knowledge. In particular, Adapter Network and Prompt Tuning methods have demonstrated great potential, achieving state-of-art results with minimal fine-tuning. Although current transfer learning strategies perform well, there is still a limitation. The Adapter Network focuses on the visual component of CLIP, while Prompt Tuning operates only on the textual component. We pose the following question. Can integrating both approaches fully leverage the potential of CLIP and further enhance generalization and performance? To explore this, we propose the method AdaptPrompt, which is introduced in the next chapter.

Chapter 3

Proposed Methodology

We will organize this chapter as follows. In Section 3.1, we present the problem formulation, outlining the objectives of our study. Section 3.2 describes related methods that we will compare in the next chapter. Section 3.3 introduces our proposed method, AdaptPrompt, providing a detailed explanation of its two core components, the adapter network and prompt tuning. Specifically, we elaborate on their design, implementation, and how they contribute to enhancing the model's adaptability and performance. Finally, Section 3.4 provides a brief summary and sets the stage for subsequent discussions.

3.1 Problem Formulation

The objective of the deepfake detection task is to discriminate between authentic images and fake images that a generative model synthesizes. The input of a deepfake detection system is the image to be discriminated, and the output should be a label of {real, fake}. The purpose of our study is to design a desired system with high performance and generalization. Formally, given an image $x \in \mathbb{R}^{H\times W\times 3}$, where H refers to the height of the input image, W refers to the width, and 3 corresponds to the number of the color channels, the goal of deepfake detection is to learn the mapping or function $f : \mathbb{R}^{H\times W\times 3} \to \{0, 1\}$, where $f(x) = 1$ if the given image is synthesized and $f(x) = 0$ otherwise.

The challenges arise from the diversity of generative models, which places high requirements on deepfake detection systems.

To address this, we proposed AdaptPrompt, a novel method that utilizes pre-trained vision-language models(VLMs) and transfer learning and fully takes advantage of the dual

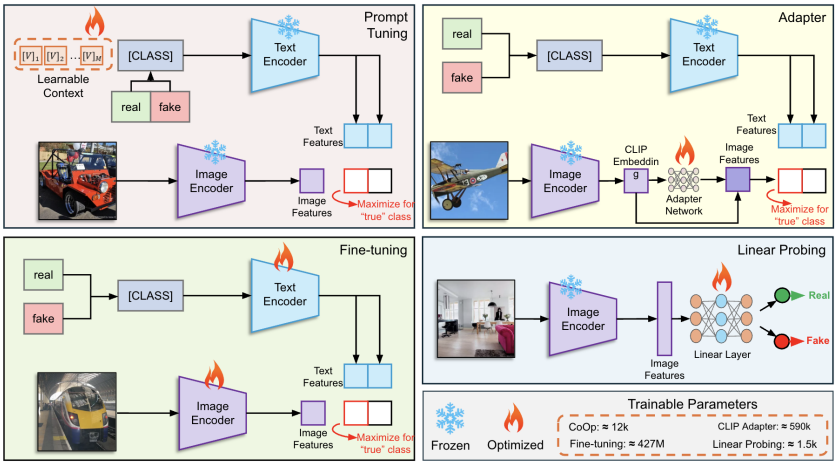


Figure 3.1: This figure showcases four transfer learning strategies for deepfake detection that were explored in the work of Khan et al.[35], including Linear Probing, Fine-tuning, Adapter Network and Prompt Tuning. At the bottom right, we annotated the number of trainable parameters.

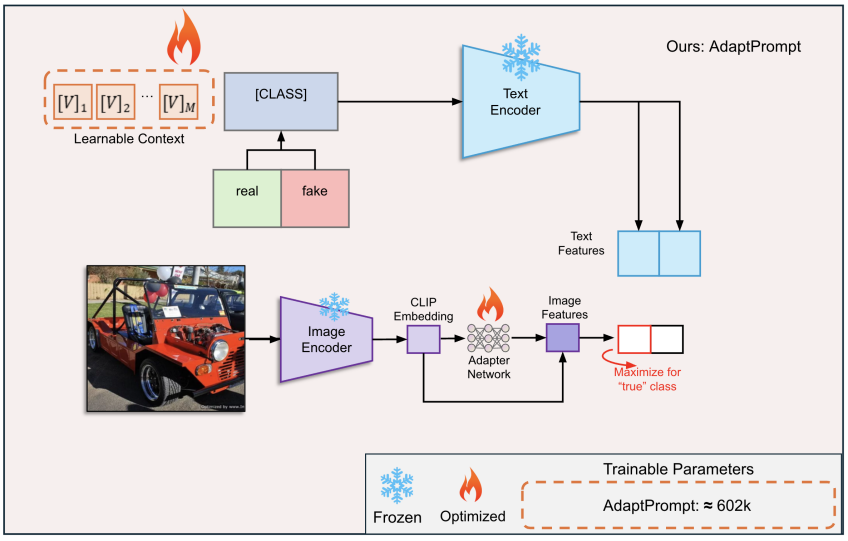


Figure 3.2: This figure showcases our proposed transfer learning strategy, AdaptPrompt. At the bottom right, we annotated the number of trainable parameters.

modality of VLMs.

3.2 Baseline

In this section, we will present the baselines that we will use in Chapter 4. These studies inspired us and led us to our proposed solution.

Wang et al.[53] explore whether a universal detector can be trained to distinguish CNN-generated images from real ones, regardless of the specific generative model or dataset used. The authors collected images from 11 different CNN-based generative models and trained ResNet-50[26] using cross-entropy loss to perform binary classification. The authors trained a classifier on ProGAN[29] and tested it on 10 other CNN-based generative models, along with discussing the effect of data augmentation. Ultimately, this method demonstrates that CNN-generated images contain detectable artifacts, making a universal detector possible. However, this work only focuses on CNN-based models, which do not include diffusion models or commercial tools. Since diffusion models and commercial tools generally produce images with higher quality and fewer detectable artifacts, this method is insufficient for real-world use.

Following this work, Gragnaniello et al.[23] modified the structure of ResNet-50[26], with the initial layers preserving high-frequency fingerprints by avoiding downsampling. During training, extensive data augmentation was applied, achieving superior performance over prior methods.

Inspired by this, Corvi et al.[11] evaluated several deepfake detection methods, respectively, trained on ProGAN[29] and Latent Diffusion[49]. The authors found that GAN-trained detectors generalize poorly to diffusion models, and training on a diffusion-based dataset improves detection for similar models.

Ojha et al.[42] found that the previous methods only detect artifacts in the images, which leads to inferior performance in unseen generators. Thus, they proposed a novel idea of using a pre-trained vision language model, CLIP[47]. By using two simple methods, linear probing and nearest neighboring, their work achieved state-of-the-art performance on both GAN-based and diffusion-based generators.

Following this, Khan et al.[35] evaluated and compared four transfer learning strategies, including Linear Probing, Fine-tuning, Adapter Network, and Prompt Tuning. Adapter Network and Prompt Tuning achieved better performance than previous methods. We present the four transfer learning strategies evaluated in this work in Figure 3.1.

3.3 AdaptPrompt Transfer Learning Strategy

3.3.1 Overview

Fine-tuning large-scale pre-trained vision-language models like CLIP is computationally expensive and often leads to overfitting, especially when the training dataset is limited. To address this, we designed AdaptPrompt, which introduces lightweight trainable modules into the pre-trained backbone while keeping the original parameters frozen. The implementation and evaluation are based on the pre-trained vision-language model CLIP. We illustrate AdaptPrompt in Figure 3.2. Compared to fine-tuning the entire model, which involves 427M parameters, AdaptPrompt maintains the integrity of the pre-trained representations while allowing task-specific adaptation through lightweight modifications, with only 602k parameters updated during the training process. This balances computational efficiency with improved generalization across unseen deepfake datasets.

3.3.2 Method Details

Adapter-Based Feature Adaptation

Adapter-based feature adaptation, originally introduced by Gao et al.[21], is to add a lightweight bottleneck layer which is optimized during the training process and keeps the rest of the model frozen. Referring to this idea, we integrate adapter layers into the CLIP backbone. Our adapter network consists of two linear layers with a non-linear activation in between. Given an input feature $X \in \mathbb{R}^{B\times 768}$, where B denotes the batch size, the transformation is defined as:

$$Y = \sigma(\sigma(XW_1^T)W_2^T)$$

where,

- $\sigma(\cdot)$ is the rectified linear unit activation function (ReLU),
- $W_1 \in \mathbb{R}^{384\times 768}$ is the weight matrix of the first linear layer,
- $W_2 \in \mathbb{R}^{768\times 384}$ is the weight matrix of the second linear layer.

The step $Y = \sigma(XW_1^T)$ projects the input to a lower-dimensional space, and the step $Z = \sigma(YW_2^T)$ projects the bottleneck features back to the original dimension. To preserve the pre-trained knowledge, we freeze all backbone parameters and train only the adapter layers, i.e., we only train about 589,824 parameters in this architecture.

Prompt Tuning for Task-Specific Adaptation

To further enhance the task-specific adaptation of the pre-trained vision-language model(VLM), we incorporate Prompt Tuning, a lightweight and efficient approach that optimizes soft prompts rather than full model parameters. Prompt Tuning is initially introduced in the domain of natural language processing[37], which involves learning randomly initialized prompts during training[56]. This method effectively optimizes the pre-trained model to focus on the target downstream tasks. In our study, we used context optimization (CoOp)[56] by Zhou et al. We construct a dynamic prompt for each class i as follows for a given downstream task with N classes and n_{ctx} context tokens:

$$P_i = [CLS] \oplus C_i \oplus [Class]$$

where

- $[CLS]$ refers to a class-specific token embedding that initializes the prompt,
- $C_i \in \mathbb{R}^{n_{ctx} \times d}$ refers to the learnable context vectors that capture task-relevant information,
- $[Class]$ refers to the tokenized class name embeddings, fixed during training, *real* or *fake* in our case,
- $\oplus$ refers to the concatenation operation along the token dimension.

These learnable prompt tokens C_i act as soft instructions, guiding the pre-trained model to attend to task-specific features without altering the model's backbone parameters.

Prediction

During training, the constructed prompt P_i is fed into the frozen CLIP text encoder, generating a task-adapted text embedding E_i:

$$E_i = TextEncoder(P_i)$$

where $TextEncoder$ is a Transformer-based model designed to convert textual descriptions into dense feature representations within a multi-modal embedding space, kept frozen during the training process in our method.

The image encoder processes the corresponding input image x to produce its feature representation F_x:

$$F_x = ImageEncoder(x)$$

where $ImageEncoder$ refers to the Vision Transformer(ViT) of CLIP, which is responsible for transforming images into feature vectors that reside in the same multi-modal embedding space as the $TextEncoder$'s outputs, kept frozen during the training process in our method.

The final classification is performed by computing the cosine similarity between the image features F_x and the class-specific text embeddings E_i:

$$s(x, i) = \frac{F_x \cdot E_i}{\parallel F_x \parallel \parallel E_i \parallel}$$

To obtain class probabilities $\hat{y}_i$, the similarity scores are passed through a softmax function:

$$\hat{y}_i = \frac{\exp(s(x, i))}{\sum_{j=1}^{N} \exp(s(x, j))}$$

where N is the number of classes, $N = 2$ in our deepfake detection task.

Optimization

The AdaptPrompt method optimizes the adapter and prompt tuning parameters using a shared cross-entropy loss. Given an input x with true label y, the model predicts the class probabilities $\hat{y}$, and the loss is defined as:

$$\mathcal{L}_{CE} = -\sum_{i} y_i \log \hat{y}_i$$

We also used the stochastic gradient descent (SGD) optimizer and involved L2 regularization when training the model to prevent overfitting. Therefore, our optimization objective is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{reg}$$

where λ is the weight decay.

Both the adapter and prompt components contribute to the final prediction, ensuring a task-specific adaptation without updating the backbone parameters.

3.4 Summary

In this section, we start with the problem formulation and then introduce the baselines, which will be used in the next chapter. Following this, we describe our proposed method, AdaptPrompt, which provides an efficient, scalable, and effective solution for deepfake detection by leveraging a pre-trained vision-language model, CLIP. Integrating adapter-based feature adaptation and prompt tuning enables task-specific learning without compromising generalization. AdaptPrompt shows the efficiency of the parameters, where only 1% of the parameters are updated in the training process. Also, the frozen backbone of the vision-language model preserves pre-trained knowledge, improving the generalization. Extensive experiments are shown in the next chapter.

Chapter 4

Experimental Results

This chapter presents the experiments conducted to evaluate our proposed methodologies. Section 4.1 provides an overview of the datasets used for both training and testing. Section 4.2 shows the results of the same set of models trained by the Diffusion Set compared to those trained by the GAN-based dataset. In Section 4.3, we show the experiments related to AdaptPrompt. Section 4.4 focuses on the experiments showing the potential of our proposed transfer learning strategy on attribution. Section 4.5 explores possibilities of improving deepfake detection performance by combining the current method with One-Shot Test-Time Training and noise prints. While these experiments did not achieve the expected improvements, we include them to document our findings and provide insights into potential limitations and future directions. Section 4.6 provides a summary of the experimental findings.

To better deliver the results, we organize our results in this section in a more concise way. Please refer to Appendix A for the original data.

4.1 Datasets

4.1.1 Training

We introduced a newly constructed dataset, Diffusion Set, to improve the generalization of deepfake detection tasks and the effectiveness of deepfake detection models on diffusion-generated images. The dataset consists of 200000 images, equally split between 100000 real images from LSUN[55] and 100000 fake images generated using diffusion models. The fake

image set is further categorized into 20 distinct object classes, with each class containing 5000 images: airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and TV monitor, which corresponds to the categories and distribution of real images.

4.1.2 Testing

To evaluate the performance and generalizability of our method, we perform extensive testing across 25 different datasets, covering a diverse range of generators. These datasets are selected to ensure a comprehensive evaluation beyond the training distribution.

The 25 testing datasets can be broadly categorized into these types: GAN-based, diffusion-based, and commercial tools. And we also include some other datasets to obtain a more comprehensive result. The detailed list is shown in Table 4.1.

4.2 Diffusion Set

The objective of this experiment is to verify whether models trained on the Diffusion Set perform better than those trained on the ProGAN Set for deepfake detection. To investigate this, we trained models using both datasets and compared their performance and generalization across four CLIP-based transfer learning strategies.

Both Diffusion Set and ProGAN Set are composed of 100k real images and 100k fake images, and the distribution across each object class is also the same. The only variation between two datasets is the method used to generate the fake images, while all other factors(e.g., dataset size, distribution across object classes, real image set) remain identical. The test set remains consistent across all experiments to ensure a fair evaluation.

4.2.1 Experimental Setup

All training images undergo the following preprocessing:

- Input size: 224x224
- Interpolation Method(for images smaller than 224x224): bicubic

Category	Generator	Num. real/fake	Real Data Source	Image Resolution
GAN	ProGAN[29]	4k/4k	LSUN	256x256
	BigGAN[5]	2k/2k	ImageNet	256x256
	CycleGAN[57]	1k/1k	Various	256x256
	EG3D[6]	1k/1k	LAION	512x512
	GauGAN[45]	5k/5k	COCO	256x256
	StarGAN[10]	2k/2k	CelebA	256x256
	StyleGAN[32]	1k/1k	LSUN	256x256
	StyleGAN2[33]	1k/1k	Various	≈256x256
	StyleGAN3[31]	≈1k/1k	Various	512x512
	Taming-T[16]	1k/1k	LAION	256x256
Diffusion	Glide_50_27[40]	1k/1k	LAION	256x256
	Glide_100_10[40]	1k/1k	LAION	256x256
	Glide_100_27[40]	1k/1k	LAION	256x256
	Guided[43]	1k/1k	LAION	256x256
	LDM_100[49]	1k/1k	LAION	256x256
	LDM_200[49]	1k/1k	LAION	256x256
	LDM_200_cfg[49]	1k/1k	LAION	256x256
	Stable Diff.[49]	1k/1k	LAION	512x512
	SDXL[46]	1k/1k	LAION	1024x1024
Comm. Tools	MidJourney-V5[39]	1k/1k	LAION	Various
	Adobe Firefly[19]	1k/1k	LAION	Various
	DALL-E 3[44]	1k/1k	LAION	Various
Other	DALL-E(mini)[13]	1k/1k	LAION	256x256
	Deepfakes[50]	≈2.7k/2.7k	YouTube	≈256x256
	FaceSwap[50]	2.8k/2.8k	YouTube	≈256x256

Table 4.1: This table presents the statistics of the test datasets. For datasets that do not include real images, we use real images from the LAION dataset as a substitute.

Method	Training Set	GAN AP/Acc	Diffusion AP/Acc	Comm. Tools AP/Acc	Average AP/Acc
Linear Probing	ProGAN	**98.22/90.92**	**95.60/86.01**	81.95/72.53	**91.92/82.85**
	Diffusion	89.57/82.97	91.68/82.98	**88.43/81.83**	89.89/82.59
Fine-tuning	ProGAN	**99.32/89.73**	97.72/92.59	98.53/94.48	**98.52/92.27**
	Diffusion	88.99/84.09	**98.35/94.23**	**99.90/98.12**	95.75/92.15
Adapter	ProGAN	**99.63/94.13**	96.83/85.70	70.29/55.17	88.92/78.33
	Diffusion	97.70/89.76	**98.02/92.20**	**99.08/95.53**	**98.27/92.50**
Prompt Tuning	ProGAN	**99.61/94.16**	97.97/86.86	85.71/59.62	94.43/80.21
	Diffusion	94.74/87.67	**97.07/90.14**	**99.22/95.62**	**97.01/91.14**

Table 4.2: This table presents the average precision and accuracy achieved by four transfer learning strategies trained using ProGAN and Diffusion Set.

- Normalization parameters: Mean: $[0.48145466, 0.4578275, 0.40821073]$, Standard Deviation: $[0.26862954, 0.26130258, 0.27577711]$
- Other data augmentation techniques include random resized crop and random flip.

To evaluate the impact of training data, we apply four CLIP-based transfer learning strategies: Linear probing, Fine-tuning, Adapter Network, and Prompt Tuning. All models utilize ViT-L/14(CLIP Backbone) as the base architecture. We optimize our network on Google COLAB T4 GPU with 180 GB RAM.

Two sets of models are evaluated on the same set of testing datasets, and the metrics include accuracy and average precision. Since the number of test sets varies across different families of generators (GAN-based, diffusion-based, commercial tools), we take the average value of each family to mitigate the bias caused by this imbalance.

4.2.2 Experimental Results

We compared scores achieved by the same set of methods trained on ProGAN and Diffusion Set in Table 4.2. As shown in the table, we can observe that the models trained on the Diffusion Set got higher scores on Diffusion-based generators and commercial tools. The difference is significant, especially for Adapter Network and Prompt Tuning methods. We also conducted experiments with smaller training set sizes. We applied stratified

Method	Num. Train Images	GAN AP/Acc	Diffusion AP/Acc	Comm.Tools AP/Acc	Average AP/Acc
Linear Probing	20k	89.92/83.26	91.78/83.13	88.19/81.70	89.96/82.70
Fine-tuning		85.43/81.81	**97.55**/**92.29**	**99.82**/**97.48**	94.27/**90.53**
Adapter		93.16/83.80	96.80/87.78	97.51/89.42	**95.82**/87.00
Prompt Tuning		**93.42**/**83.74**	95.26/86.49	98.39/88.59	95.69/86.27
Linear Probing	40k	89.05/81.82	91.34/82.69	87.42/81.53	89.27/82.01
Fine-tuning		55.98/53.83	91.74/83.81	98.60/**94.97**	82.11/77.54
Adapter		**94.09**/**85.66**	**97.31**/**91.13**	98.64/94.70	**96.68**/**90.50**
Prompt Tuning		91.79/84.12	94.85/87.93	**99.25**/93.30	95.30/88.45
Linear Probing	60k	89.33/82.33	91.77/83.48	87.89/81.92	89.66/82.58
Fine-tuning		82.96/74.48	**98.25**/**93.51**	**99.94**/**96.70**	93.72/88.23
Adapter		**95.83**/**87.34**	97.61/91.72	98.61/94.53	**97.35**/**91.20**
Prompt Tuning		93.69/86.07	95.28/88.25	99.38/94.45	96.12/89.59
Linear Probing	80k	89.74/82.85	92.15/83.72	88.39/82.05	90.09/82.87
Fine-tuning		86.59/83.59	96.16/**89.00**	**99.63**/**97.32**	94.13/**89.97**
Adapter		**96.44**/**87.97**	**97.56**/88.70	98.80/92.58	**97.60**/89.75
Prompt Tuning		90.25/82.25	95.78/88.73	98.21/93.32	94.75/88.10

Table 4.3: This table showcases the scores achieved by the methods trained using a smaller-sized Diffusion Set. Results are organized based on the total number of training images: 20k, 40k, 60k, and 80k. Each dataset consists of an equal proportion of real and fake images; e.g. for a 20k dataset, we have 10k real and 10k fake images.

subsampling to maintain the original class distribution and formed four smaller datasets containing 20k, 40k, 60k, and 80k images, respectively. In each dataset, we ensured that the ratio of real to fake images remained unchanged, as well as the proportion of each object category. As shown in Table 4.3, generally, the larger training datasets yield higher scores, especially for adapter and prompt tuning methods. The differences in performance brought by different training set sizes are not significant.

4.3 AdaptPrompt Experiments

4.3.1 Experimental Setup

In this section, we show the experimental results related to our proposed method and two modified versions. For AdaptPrompt without a projection layer, we removed the projection layer of CLIP from the model. The removal of the projection layer eliminates the final mapping of feature representations into the designated embedding space, which may preserve more raw information. And for AdaptPrompt without the last transformer block, we removed the last transformer block of CLIP. The removal of the last transformer block eliminates the final refinement of feature representations and aggregation of global context. Thus, the model relies on the outputs from the penultimate layer, which may lead to less polished embeddings but can potentially reduce overfitting and lower model complexity. All the tables that showcase the results will include three versions of AdaptPrompt to provide an intuitive view. Then, we compare our results with the other methods trained on the identical training set (Diffusion Set) with the identical setting. We follow the same experimental setup as outlined in Section 4.2.1.

4.3.2 Experimental Results

The results are shown in Table 4.4.

Trained on the same training set with the same setting, AdaptPrompt with the last Transformer block removed consistently performs better than the other two versions of AdaptPrompt. While AdaptPrompt with the projection layer removed did not show any difference with the original version of AdaptPrompt. As the only architectural difference is removing the last Transformer block, such modification may improve the model's adaptability to the task, which may extend to other tasks relying on large-scale pre-trained models.

Beyond that, AdaptPrompt, with the last transformer block removed, outperforms any other methods listed regarding the mean Average Precision and Accuracy. Though fine-tuning gives the best performance for diffusion-based generators and commercial tools, it exhibits instability for GAN-based generators. In contrast, AdaptPrompt with the last transformer block removed demonstrates stronger generalization ability, maintaining consistently high performance even on datasets where the test sets differ significantly from the training set. This suggests that AdaptPrompt is more adaptable to unseen data, likely due

Method	GAN AP/Acc	Diffusion AP/Acc	Comm.Tools AP/Acc	Average AP/Acc
Wang et al.[53]	92.32/78.23	74.29/ 57.15	61.57/ 52.43	76.06/ 62.60
Gragn et al.[23]	98.88/ **92.83**	92.86/ 64.53	72.58/ 56.53	88.11 /71.30
Corvi et al.[11]	**99.14**/ 90.67	86.06/ 57.15	66.40/ 54.62	83.87/ 67.48
Ojha et al.[42]	95.39/86.13	93.66 /82.77	75.26 /68.42	88.10/ 79.11
Linear Probing[35]	89.57/82.97	91.68/82.98	88.43/81.83	89.89/82.59
Fine-tuning[35]	88.99/84.09	**98.35**/**94.23**	**99.90**/**98.12**	95.75/92.15
Adapter[35]	97.70/89.76	98.02/92.20	99.08/95.53	98.27/92.50
Prompt Tuning[35]	94.74/87.67	97.07/90.14	99.22/95.62	97.01/91.14
AdaptPrompt_v0(Ours)	96.39/87.49	97.04/91.24	98.78/94.68	97.40/91.14
AdaptPrompt_v1(Ours)	96.39/87.49	97.04/91.24	98.78/94.68	97.40/91.14
AdaptPrompt_v2(Ours)	97.98/88.33	98.33/93.18	99.48/96.65	**98.60**/**92.72**

Table 4.4: This table showcases the performance of five transfer learning strategies trained on the complete Diffusion Set (including 100k real images and 100k fake images). AdaptPrompt_v0 represents the original version of AdaptPrompt, AdaptPrompt_v1 represents AdaptPrompt without the projection layer, and AdaptPrompt_v2 represents AdaptPrompt without the last transformer block.

to its ability to retain pre-trained knowledge while efficiently adapting to new data distributions. Instead of fully modifying the backbones as in fine-tuning, AdaptPrompt leverages prompt-based and adapter-based adjustments, allowing it to capture new patterns while preserving generalizable representations.

We also conducted experiments on smaller training sets to investigate the performance and generalization of AdaptPrompt when the size of the training set is limited. As shown in Figure 4.5, the performance differences among the three versions of AdaptPrompt are minimal, with AdaptPrompt with the last transformer block removed slightly outperforming the other two. Moreover, all versions maintain a consistently high level of performance across the dataset. We further conducted a few-shot analysis, as shown in Figure 4.6. Even with only 640 training images (320 real and 320 fake), the model still achieves an accuracy of over 70%.

Method	Num. Train Images	GAN AP/Acc	Diffusion AP/Acc	Comm. Tools AP/Acc	Average AP/Acc
AdaptPrompt_v0	20k	93.81/**86.33**	**97.98**/**91.96**	98.03/91.90	96.61/90.06
AdaptPrompt_v1		91.85/84.28	97.06/90.77	98.09/93.13	95.67/89.39
AdaptPrompt_v2		**96.98**/85.28	97.12/90.97	**99.04**/**94.90**	**97.71**/**90.38**
AdaptPrompt_v0	40k	95.64/88.31	**97.88**/91.03	**99.02**/**94.85**	97.51/**91.40**
AdaptPrompt_v1		95.19/84.95	97.64/**92.06**	98.83/94.22	97.22/90.41
AdaptPrompt_v2		**97.61**/**89.60**	97.31/85.28	98.93/91.72	**97.95**/88.87
AdaptPrompt_v0	60k	96.97/86.94	97.62/91.97	98.81/94.45	**97.80**/**91.12**
AdaptPrompt_v1		95.80/85.12	**98.10**/**92.65**	**99.38**/**94.78**	97.76/90.85
AdaptPrompt_v2		**98.01**/**88.52**	91.82/80.75	99.37/88.67	96.40/85.98
AdaptPrompt_v0	80k	94.04/86.70	**98.19**/**91.69**	**99.40**/95.73	97.21/91.37
AdaptPrompt_v1		96.46/88.28	97.35/91.42	98.78/94.60	97.53/91.43
AdaptPrompt_v2		**97.10**/**89.09**	97.82/90.81	99.19/**95.78**	**98.04**/**91.89**

Table 4.5: This table shows the performance scores of AdaptPrompt trained on 20k, 40k, 60k, 80k dataset.

Method	GAN AP/Acc	Diffusion AP/Acc	Comm. Tools AP/Acc	Average AP/Acc
AdaptPrompt_v0	77.88/71.55	84.19/76.79	81.93/74.53	81.33/74.29
AdaptPrompt_v1	77.23/72.25	**85.72**/**78.07**	83.01/75.40	81.99/75.24
AdaptPrompt_v2	**84.76**/**76.35**	84.60/76.84	**87.84**/**80.22**	**85.74**/**77.80**

Table 4.6: This table presents the results from our few-shot (32-shot) experiment.

Method	GAN AP/Acc	Diffusion AP/Acc	Comm.Tools AP/Acc	Average AP/Acc
Fixed Prompt	**98.26/90.06**	97.86/87.92	99.62/95.90	98.58/91.29
Learnable Prompt	97.98/88.33	**98.33/93.18**	**99.48/96.65**	**98.60/92.72**

Table 4.7: This table shows the scores achieved by fixed prompt AdaptPrompt and learnable prompt AdaptPrompt.

4.3.3 Ablation Studies

Effect of Prompt Tuning

We evaluated whether AdaptPrompt benefits from fixed prompts (predefined and manually designed) or learnable prompts (optimized during training). The goal is to determine if AdaptPrompt's effectiveness comes primarily from its structural design or from its ability to dynamically adjust prompt representations. The baseline we used here is AdaptPrompt, with the last Transformer block removed.

To evaluate, we compare two variations of AdaptPrompt:

- Fixed Prompt AdaptPrompt: A manually predefined prompt remains constant throughout training. e.g. for a fake image, the prompt would be *a fake photo*.
- Learnable Prompt AdaptPrompt: The model optimizes the prompt as trainable parameters, we set the length of prompt as 16.

As shown in Table 4.7, we observe that learnable prompt yields higher scores regarding the performance on diffusion-based, commercial tools, and average scores. To investigate the effect of prompts further, we then set more complicated prompts, for fixed prompt AdaptPrompt, adding the information of generators or the name of the object, while the performance further degraded.

Effect of Adapter Network

To evaluate the contribution of the adapter network in AdaptPrompt, we remove the adapter network to compare. We still used the AdaptPrompt with the last Transformer block removed as the baseline here. As shown in Table 4.8, AdaptPrompt outperforms the prompt-only method on all the tests.

Method	GAN AP/Acc	Diffusion AP/Acc	Comm.Tools AP/Acc	Average AP/Acc
Prompt-only	96.41/87.92	96.70/90.30	99.27/95.25	97.46/91.16
AdaptPrompt	**97.98/88.33**	**98.33/93.18**	**99.48/96.65**	**98.60/92.72**

Table 4.8: Performance comparison between prompt-only AdaptPrompt and full AdaptPrompt.

4.4 Attribution

In this section, we introduced our experimental setup and results investigating the effectiveness of AdaptPrompt in the deepfake attribution task, which involves identifying the specific generator or the family of generators responsible for creating a given fake image. Though the test set size is limited, we aim to investigate if AdaptPrompt has the potential to perform attribution tasks.

4.4.1 Experimental Setup

In this experiment, we modified our code to perform multi-class classification instead of binary. We use the test datasets in three main generator families (GAN-based, diffusion-based, and commercial tools) listed in Table 4.1 as the training set. From each dataset, we randomly sample 640 fake images to form a new training dataset. Then, we train AdaptPrompt to perform classification among 22 datasets. We also utilized the rest of the fake images to test the performance. As most datasets we have only contain 1000 images, the result we have would based on 360 images, and this might lead to biased results.

4.4.2 Experimental Results

Figure 4.1 shows the confusion matrix for the deepfake attribution task, with darker blue indicating higher accuracy. Overall, the model achieves an accuracy of 81.4% in correctly predicting the label. Additionally, 96.49% of the predictions fall within the correct family of generators, indicating that while the model can effectively distinguish between different generator families, it struggles with finer-grained attribution within the same family, especially the different datasets generated by the same generator. This might indicate that different configurations do not introduce additional detectable artifacts, meaning that we

may attribute an image to a specific generator but not to a particular configuration within it. Despite these challenges, the transfer learning strategy designed for deepfake detection also demonstrates strong potential for deepfake attribution, as it consistently identifies the generator family and shows promising results. This could be a new direction for our future work.

4.5 Unsuccessful Attempts

4.5.1 Experiments on Combination with One-Shot Test-Time Training

We attempted to combine transfer learning with one-shot test-time training[8]. In this approach, during testing, we sample a few images from the test set whose authenticity is unknown and use faceswap techniques to replace the facial region, generating images that are definitively fake. These generated fake images are then used for one-shot training. This method aims to reduce the gap between the training and testing datasets. However, in our experiments, we found that this approach is only effective for fake images containing frontal human faces. We encountered difficulties adapting this method to images without human faces, which limiting its broader applicability. Our approach for images without human faces involved randomly cropping a rectangular region from one image, pasting it onto a random location in another image, and blending the edges to create a seamless transition. However, this method did not improve the model performance. The adaptation of this method may require further investigation in order to create new fakes.

4.5.2 Experiments on Combination with NoisePrint and TruFor

We tried to extract the noise prints from the images and learn patterns within the noise print to distinguish between real and fake images. Our approach to extracting noise prints refers to NoisePrint[12] and TruFor[24]. When integrating this approach into our model, we modified the original input format. Our model takes 3-channel inputs that correspond to the RGB representation of an image. We experimented with modifying the number of channels so that the model could learn both the noise print and the image features simultaneously or learn the noise print only. However, neither approach led to performance improvements.

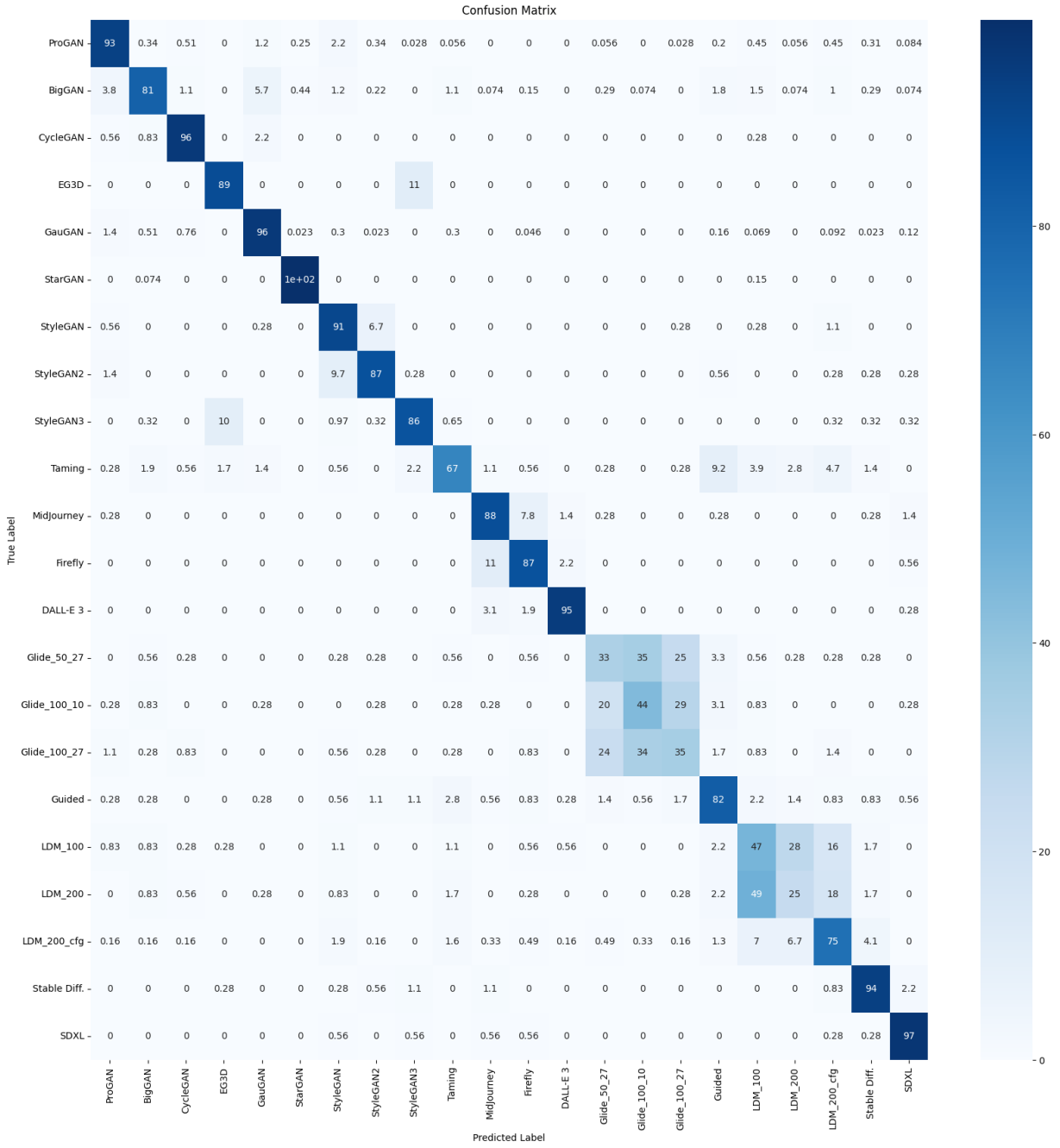


Figure 4.1: Confusion matrix for the deepfake attribution task, where each entry represents the classification percentage for a given generator. The vertical axis (rows) corresponds to actual generator labels, and the horizontal axis (columns) corresponds to the predicted labels.

In addition, training and inference time increased significantly. In our original architecture, few-shot training on 640 images took less than one minute per epoch. After incorporating the noise print extraction step, the processing time increased to 4 h per epoch. Despite optimizing the code structure and reducing the run-time to 45 minutes per epoch, overall performance remained unsatisfactory. Thus, we have decided to discontinue this approach.

4.6 Summary

In this chapter, we conducted extensive experiments to evaluate the Diffusion Set dataset and the effectiveness and generalization capability of AdaptPrompt for deepfake detection. Our findings demonstrate that the proposed method outperforms existing transfer learning strategies, particularly in handling diffusion-based and commercial tool-generated images. The key findings are summarized as follows. We introduced Diffsion Set, a dataset specifically designed to improve the generalization of deepfake detection models. Models trained on the Diffusion Set exhibited superior performance compared to those trained on traditional GAN-based datasets. We then evaluated four transfer learning strategies (linear probing, fine tuning, adapter network, and prompt tuning). Among them, AdaptPrompt achieved the best generalization, especially when applied to previously unseen generative models. The experiment shows that AdaptPrompt remains in good performance even when the size of the training set is limited and demonstrates a strong potential for deepfake attribution, achieving 81.4% accuracy in generator classification and correctly identifying the generator family in 96. 49% cases. In general, our results confirm that AdaptPrompt provides an effective and scalable solution for deepfake detection using vision-language models and transfer learning techniques.

Chapter 5

Conclusion

In this thesis, we discuss the existing literature work in the field of deepfake detection. Traditional detection methods often struggle to generalize across unseen generative models, necessitating new approaches that enhance generalization capability.

In this thesis, we introduced AdaptPrompt, a novel transfer learning strategy that integrates adapter networks and prompt tuning to effectively adapt pre-trained vision-language models(VLMs) for deepfake detection. Unlike conventional fine-tuning approaches, AdaptPrompt maintains the pre-trained model's frozen backbone while introducing lightweight, trainable modules. This design significantly improves generalization, computational efficiency, and adaptability in different deepfake generation methods. We also constructed Diffusion Set, a dataset specifically designed to improve the generalization of deepfake detection models, and conducted extensive experiments across 25 test datasets. Our findings confirm that AdaptPrompt outperforms existing methods in terms of generalization, making it a strong candidate for real-world deepfake detection tasks.

The experimental results validate the effectiveness of AdaptPrompt and highlight several key findings. First, we evaluated the diffusion set using four transfer learning strategies: linear probing, fine-tuning, adapter network, and prompt tuning. We found that the models trained on the Diffusion Set outperform the same set of models trained on the traditional GAN-based training set, particularly when tested on unseen diffusion-generated images. Also, AdaptPrompt exhibits strong cross-model generalization, maintaining high accuracy even on the test sets from another generator family that significantly differ from the training data. Furthermore, AdaptPrompt maintains over 70% accuracy even when trained on just 640 images (320 real, 320 fake), demonstrating its effectiveness in low-data scenarios. Lastly, AdaptPrompt achieves 81.4% accuracy in attributing fake images to spe-

cific generative and 96.49% accuracy in correctly identifying generator families, indicating its strong potential for attribution tasks.

To further enhance AdaptPrompt and its applicability to deepfake detection and related tasks, possible directions for future research may include the following ones.

- Expanding transfer learning strategies: Investigate additional parameter-efficient tuning methods, such as low-rank adaptation(LoRA).
- Further investigating the improvements on the model structure: In the thesis, we compared the performance of the original model and the one with the last transformer block removed, and the results suggest the latter outperforms the former. We may need to conduct more experiments to investigate if similar modifications contribute to minimizing overfitting and, therefore, improving the performance of the downstream task.
- Open-set attribution: The current experiments on attribution are closed-set attribution, which means all possible test images are included in the training set. One possible direction for future research would be to perform clustering instead of classification with the same transfer learning strategy, enabling the model to detect fake images generated by previously unseen generative models.

References

[1] Deepfacelab. https://github.com/iperov/deepfacelab.

[2] Dfaker. https://github.com/dfaker/df.

[3] Adobe. Adjust and exaggerate facial features, 2022.

[4] Charlotte Bird, Eddie L. Ungless, and Atoosa Kasirzadeh. Typology of risks of generative text-to-image models, 2023.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2018.

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, 2022.

[7] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18710–18719, 2022.

[8] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. OST: Improving generalization of deepfake detection via one-shot test-time training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[9] Zhiyi Chen, Jinyi Ye, Emilio Ferrara, and Luca Luceri. Prevalence, sharing patterns, and spreaders of multimodal ai-generated content on x during the 2024 u.s. presidential election, 2025.

[10] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797, 2018.

[11] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[12] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020.

[13] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phuć Lê Khàc, Luke Melas, and Ritobrata Ghosh. Dall·e mini, 2021.

[14] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7890–7899, 2020.

[15] Tarik Dzanic, Karan Shah, and Freddie D. Witherden. Fourier spectrum discrepancies in deep network generated images. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3022–3032, 2020.

[16] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, 2021.

[17] Hany Farid. Lighting (in)consistency of paint by text. *arXiv preprint*, arXiv:2207.13744v2, 2022.

[18] Hany Farid. Perspective (in)consistency of paint by text. *arXiv preprint*, arXiv:2206.14617v1, 2022.

[19] Adobe Firefly. Adobe firefly, 2023.

[20] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3247–3258, 2020.

[21] 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*, pages 1–15, 2023.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[23] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[24] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization, 2023.

[25] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

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

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint*, arXiv:2106.09685, 2021.

[28] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, pages 4904–4916. PMLR, 2021.

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[30] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:12104–12114, 2020.

[31] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:852–863, 2021.

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator ar- chitecture for generative adversarial networks. In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[33] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020.

[34] Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. Deepfake detection: Analysing model generalisation across architectures, datasets and pre-training paradigms. *IEEE Access*, 2023.

[35] Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. Clipping the deception: Adapting vision-language models for universal deepfake detection, 2024.

[36] Konwoo Kim, Michael Laskin, Igor Mordatch, and Deepak Pathak. How to adapt your large-scale vision-and-language model, 2022.

[37] 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*, 55(9):1–35, 2023.

[38] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACV Workshops)*, pages 83–92, 2019.

[39] Midjourney. Midjourney, 2023.

[40] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint*, arXiv:2112.10741, 2021.

[41] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, pages 2642–2651. PMLR, 2017.

[42] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24480–24489, 2023.

[43] OpenAI. Guided-diffusion, 2021.

[44] OpenAI. Dall·e 3, 2023.

[45] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019.

[46] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint*, arXiv:2307.01952, 2023.

[47] 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 *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint*, arXiv:2204.06125, 2022.

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[50] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, 2019.

[51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[52] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, pages 2256–2265. PMLR, 2015.

[53] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8695–8704, 2020.

[54] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018.

[55] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint*, arXiv:1506.03365, 2015.

[56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017.

[58] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z. Li. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2929–2939, 2021.

APPENDICES

Appendix A

Original Data

In Appendix A, we present the original dataset used in our study. The sections labeled as "Ours" indicate results obtained from our own experiments, while the performance of other models on various test datasets has been extracted from existing literature. For clarity and transparency, we have annotated the sources of all externally referenced data. We evaluated the LDM and Glide models using three different configurations, generating three distinct datasets for each. Due to space constraints, the table presents the average results across these three datasets in the corresponding entries. The final averaged values at the end of the list are computed directly from the original data rather than derived from the previously averaged results.

"Ours Version 0" refers to the methods with original structure trained on Diffusion Set, "Ours Version 1" refers to the methods with projection layer removed trained on Diffusion Set, and "Ours Version 2" refers to the methods with the last transformer block removed traiend on Diffusion Set.

Table A.1: Original data on average precision.

Method	Variant	Generative Adversarial Networks										DALL-E(mini)	Denoising Diffusion Models					Commercial Tools			FF++		mAP
		Pro GAN	Big GAN	Cycle GAN	EG3D	Gau GAN	Star GAN	Style GAN	Style GAN-2	Style GAN-3	Taming-T		Glide	Guided	LDM	SD	SDXL	DALL-E3	MidJourney	Firefly	Deep Fakes	Face Swap	
Wang et al.[53] (CVPR'20)	Blur+JPEG (0.1)	100.00	83.04	90.09	95.58	88.94	97.18	99.27	96.43	98.63	73.9	67.47	81.02	83.1	68.61	64.33	72.27	-	-	-	75.88	50.78	81.18
	Blur+JPEG (0.5)	100.00	82.63	94.71	55.32	96.62	93.88	93.25	88.64	85.33	59.78	60.92	69.75	65.11	60.24	52.14	65.92	-	-	-	64.33	49.76	72.65
Gragn. et al.[23] (ICME'21)	ResNet-50 No Downsample	100.00	97.57	97.63	99.95	98.36	99.99	100.00	99.98	100.00	95.31	91.32	94.08	93.81	92.33	91.75	90.93	-	-	-	95.90	61.54	94.24
Corvi et al.[11] (ICASSP'23)	ProGAN/LSUN	100.00	99.66	97.94	99.92	99.74	99.95	100.00	99.96	99.93	94.34	95.45	89.51	79.30	88.26	87.01	74.90	-	-	-	95.52	56.58	91.52
	Latent/LSUN	91.83	74.25	49.05	42.87	89.14	50.19	73.25	74.73	70.20	95.21	98.15	87.35	59.17	100.00	100.00	99.23	-	-	-	83.70	45.52	79.93
Ojha et al.[42] (CVPR'23)	CLIP Linear Probing	99.99	98.73	98.92	79.58	99.74	96.06	95.73	95.81	92.21	97.12	96.84	93.85	92.09	95.71	93.58	88.55	-	-	-	77.48	75.87	93.05
Khan. et al.[35] (ICMR'24)	Linear Probing	99.91	97.77	98.53	99.48	99.69	99.00	95.53	94.98	99.54	97.74	95.65	97.75	92.14	95.94	92.24	94.99	-	-	-	80.07	76.58	95.22
	Fine Tuning	100.00	98.65	99.00	99.97	98.12	100.00	99.61	99.48	100.00	98.38	98.15	96.23	97.40	98.79	97.53	99.52	-	-	-	87.42	60.22	96.29
	Adapter	100.00	99.58	99.97	99.50	99.98	99.98	99.44	98.80	99.83	99.27	98.60	99.26	96.16	97.76	91.90	92.32	-	-	-	91.37	82.11	97.27
	Prompt Tuning	100.00	99.42	99.92	99.51	99.95	99.97	99.52	98.62	99.68	99.54	98.89	99.32	97.41	97.91	96.23	96.42	-	-	-	92.59	88.01	98.06
Ours - Version 0	Linear Probing	97.64	74.51	71.70	98.00	71.56	89.04	99.18	95.14	99.72	99.22	97.98	83.45	96.19	92.97	99.89	99.83	68.12	99.05	98.12	47.65	55.70	87.50
	Fine Tuning	99.90	70.93	60.61	100.00	61.54	99.72	98.76	98.79	100.00	99.62	99.81	95.89	97.93	99.86	99.97	99.95	99.88	99.90	99.93	63.09	55.57	91.73
	Adapter	99.90	94.87	91.51	99.97	95.41	99.10	99.20	97.51	99.95	99.54	99.43	97.01	97.4	98.01	99.96	99.80	98.86	98.69	99.70	66.17	66.39	95.53
	Prompt Tuning	99.79	88.53	79.81	99.97	85.35	98.46	99.29	97.08	99.81	99.32	99.37	95.22	96.59	97.26	99.86	99.71	99.09	98.84	99.74	61.07	61.3	93.62
	AdaptPrompt	99.87	91.53	89.59	99.95	89.69	98.69	98.91	96.84	99.72	99.09	99.08	95.15	96.04	97.71	99.98	99.73	98.66	98.17	99.52	63.95	65.01	94.50
Ours - Version 1	Fine Tuning	99.41	65.5	52.75	100.00	65.88	99.48	97.93	97.96	100.00	99.52	99.16	93.43	97.23	99.71	100.00	99.98	99.98	99.84	99.82	54.37	52.73	90.44
	Adapter	99.90	94.86	91.50	99.97	95.41	99.09	99.2	97.51	99.95	99.54	99.44	97.01	97.39	98.01	99.96	99.8	98.87	98.69	99.70	66.16	66.37	95.53
	Prompt Tuning	99.79	88.53	79.81	99.97	85.35	98.46	99.29	97.08	99.81	99.32	99.37	95.22	96.59	97.26	99.86	99.71	99.09	98.84	99.74	61.07	61.30	93.62
	AdaptPrompt	99.87	91.53	89.59	99.95	89.69	98.69	98.91	96.84	99.72	99.09	99.08	95.15	96.04	97.38	99.98	99.73	98.66	98.17	99.52	63.95	65.01	94.46
Ours - Version 2	Linear Probing	98.53	83.06	75.70	96.59	79.07	95.88	98.99	95.01	99.56	99.25	99.81	85.50	95.98	92.76	99.85	99.78	76.72	99.07	98.87	63.09	55.57	89.81
	Fine Tuning	99.77	70.84	59.23	100.00	62.49	99.02	97.87	98.50	100.00	99.56	99.55	95.77	97.35	99.75	99.99	99.94	99.94	99.78	99.75	62.06	53.54	91.43
	Adapter	99.76	95.64	95.93	99.99	97.00	99.76	98.89	96.26	99.89	99.50	99.23	96.70	97.52	97.74	99.99	99.91	99.54	99.40	99.93	69.66	69.35	96.02
	Prompt Tuning	99.78	90.29	89.93	99.85	90.70	99.80	99.17	95.91	99.85	98.77	98.47	95.10	95.56	96.72	99.74	99.52	99.13	98.98	99.70	55.98	60.32	93.88
	AdaptPrompt	99.82	94.24	95.46	99.99	95.50	99.78	99.01	96.41	99.95	99.68	99.33	97.66	98.24	97.95	99.98	99.90	99.30	99.25	99.88	67.06	68.17	95.91

Table A.2: Original data on accuracy.

Method	Variant	Generative Adversarial Networks										DALL-E(mini)	Denoising Diffusion Models					Commercial Tools			FF++		Avg. Acc
		Pro GAN	Big GAN	Cycle GAN	EG3D	Gau GAN	Star GAN	Style GAN	Style GAN-2	Style GAN-3	Taming-T		Glide	Guided	LDM	SD	SDXL	DALL-E3	MidJourney	Firefly	Deep Fakes	Face Swap	
Wang et al.[53] (CVPR'20)	Blur+JPEG (0.1)	99.90	67.65	79.50	72.65	76.63	89.72	82.10	77.05	80.68	56.45	55.05	61.15	62.90	54.03	52.50	53.40	-	-	-	52.67	49.68	66.09
	Blur+JPEG (0.5)	99.65	58.13	77.80	50.30	75.56	79.99	69.80	62.30	53.42	51.05	51.90	54.33	52.35	51.35	50.15	51.00	-	-	-	51.46	50.02	59.18
Gragn. et al.[23] (ICME'21)	ResNet-50 No Downsample	100.00	93.27	91.75	97.55	94.13	99.65	97.25	89.75	97.47	67.45	60.65	69.38	67.3	62.33	59.70	57.75	-	-	-	65.31	50.02	76.59
Corvi et al.[11] (ICASSP'23)	ProGAN/LSUN	100.00	95.85	90.35	98.40	92.46	99.00	97.65	84.90	82.79	65.30	69.30	58.98	53.10	58.83	55.70	52.10	-	-	-	59.38	50.11	72.72
	Latent/LSUN	50.94	51.82	46.20	49.25	50.86	48.02	59.40	50.95	50.05	77.65	87.00	59.83	50.95	99.25	99.25	93.10	-	-	-	69.87	48.14	66.40
Ojha et al.[42] (CVPR'23)	CLIP Linear Probing	98.94	94.48	94.20	57.75	94.65	87.49	85.55	83.40	75.42	89.45	89.20	82.15	79.00	87.80	81.90	74.15	-	-	-	62.71	64.30	82.84
Khan. et al.[35] (ICMR'24)	Linear Probing	98.50	91.75	91.00	98.20	88.08	94.42	81.40	71.70	94.11	91.05	85.80	90.55	79.05	87.42	77.30	83.85	-	-	-	69.37	68.30	86.26
	Fine Tuning	99.60	77.38	71.55	98.40	65.70	100.00	94.85	95.30	99.89	94.40	93.20	88.78	92.35	95.17	91.75	97.35	-	-	-	76.46	52.11	88.74
	Adapter	99.88	94.75	97.45	95.30	95.47	99.12	93.35	78.35	93.11	94.55	92.00	94.27	81.65	89.18	67.70	71.60	-	-	-	77.11	70.16	88.72
	Prompt Tuning	99.83	93.80	95.60	93.50	93.43	99.15	95.25	82.95	93.11	94.95	91.50	92.88	84.30	88.16	76.45	77.80	-	-	-	78.45	74.66	89.45
Ours Version 0	Linear Probing	89.14	64.95	66.85	95.10	60.13	79.83	95.15	86.70	97.58	94.15	92.65	71.57	89.45	84.07	95.25	95.25	58.25	94.00	93.25	49.02	51.04	80.59
	Fine Tuning	88.79	65.12	63.50	98.50	53.78	96.82	87.25	89.80	99.68	97.65	97.70	87.55	94.50	98.05	98.40	98.35	98.15	98.05	98.15	52.43	51.21	87.39
	Adapter	98.24	73.95	79.40	98.35	74.52	95.42	95.20	86.75	98.89	96.85	96.45	88.68	91.00	92.18	98.35	97.85	94.70	94.25	97.65	54.45	57.11	88.88
	Prompt Tuning	93.76	73.65	73.55	97.60	72.09	90.40	95.05	86.30	98.84	95.50	95.65	84.98	89.85	90.62	97.50	97.10	95.35	94.55	96.95	51.58	52.00	86.96
	AdaptPrompt	98.40	69.72	71.35	96.20	66.65	93.87	95.35	91.00	97.84	94.55	94.65	88.18	89.20	91.75	96.20	96.00	94.55	93.50	96.00	53.63	54.95	87.34
Ours Version 1	Fine Tuning	92.51	61.23	61.10	94.55	51.83	96.17	88.75	90.40	98.05	93.85	93.05	85.23	90.60	94.28	94.55	94.55	94.55	94.30	94.30	50.19	51.32	84.98
	Adapter	98.24	73.95	79.35	98.35	74.50	95.42	95.20	86.75	98.89	96.85	96.45	88.71	91.00	92.20	98.35	97.85	94.75	94.25	97.65	54.45	57.11	88.88
	Prompt Tuning	93.76	73.65	73.55	97.60	72.09	90.40	95.05	86.30	98.84	95.50	95.65	84.98	89.85	90.62	97.50	97.10	95.35	94.55	96.95	51.58	52.00	86.96
	AdaptPrompt	98.40	69.72	71.35	96.20	66.65	93.87	95.35	91.00	97.84	94.55	94.65	88.18	89.20	91.75	96.20	96.00	94.55	93.50	96.00	53.63	54.95	87.34
Ours Version 2	Linear Probing	93.24	68.58	65.50	93.95	64.83	87.97	93.70	86.70	96.37	94.80	97.70	72.88	89.05	83.52	95.90	95.80	64.00	94.45	94.70	52.43	51.21	82.00
	Fine Tuning	92.11	64.78	65.55	98.10	53.57	94.20	87.70	87.75	99.26	97.10	96.75	88.23	92.95	97.32	98.10	97.80	97.90	97.15	96.90	50.73	52.00	87.08
	Adapter	97.45	80.67	80.90	99.10	79.98	97.72	90.40	79.45	99.05	95.85	93.50	81.63	85.75	87.73	99.15	98.25	94.25	94.80	98.65	55.00	54.82	87.31
	Prompt Tuning	98.20	70.15	72.90	96.15	64.22	97.85	95.20	90.40	99.85	94.25	93.65	87.05	88.90	90.27	96.15	95.70	95.00	94.75	96.00	52.30	54.12	87.11
	AdaptPrompt	98.46	70.47	70.10	98.10	69.99	97.72	94.35	89.80	97.26	97.00	95.75	90.70	93.50	92.33	98.10	97.90	96.00	96.10	97.85	57.51	58.18	88.93