GENERALIZABLE FACE FORGERY DETECTION WITH METRIC LEARNING AND DOMAIN-ADVERSARIAL TRAINING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Mustafa Hakan Kara
April 2025

GENERALIZABLE FACE FORGERY DETECTION WITH METRIC LEARNING AND DOMAIN-ADVERSARIAL TRAINING
By Mustafa Hakan Kara
April 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Uğur Güdükbay (Advisor)

Ayşegül Dündar Boral (Co-Advisor)

Özgür Ulusoy

Yusuf Sahillioğlu (METU)

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

GENERALIZABLE FACE FORGERY DETECTION WITH METRIC LEARNING AND DOMAIN-ADVERSARIAL TRAINING

Mustafa Hakan Kara
M.S. in Computer Engineering
Advisor: Uğur Güdükbay
Co-Advisor: Ayşegül Dündar Boral
April 2025

As face forgeries generated by deep neural networks become increasingly sophisticated, detecting face manipulations in digital media has posed a significant challenge, underscoring the importance of maintaining digital media integrity and combating visual disinformation. Current detection models, predominantly based on supervised training with domain-specific data, often falter against forgeries generated by unencountered techniques. In response to this challenge, we introduce *Trident*, a face forgery detection framework that employs triplet learning with a Siamese network architecture for enhanced adaptability across diverse forgery methods. *Trident* is trained on curated triplets to isolate nuanced differences of forgeries, capturing fine-grained features that distinguish pristine samples from manipulated ones while controlling for other variables. To further enhance generalizability, we incorporate domain-adversarial training with a Forgery Discriminator. This adversarial component guides our embedding model towards forgery-agnostic representations, improving its robustness to unseen manipulations. In addition, we prevent gradient flow from the classifier head to the embedding model, avoiding overfitting induced by artifacts peculiar to certain forgeries. Comprehensive evaluations across multiple benchmarks and ablation studies demonstrate the effectiveness of our framework.

Keywords: Deepfake detection, metric learning, triplet learning, domain-adversarial training, face forgery detection.

ÖZET

METRİK ÖĞRENME VE ALAN-ÇEKİŞMELİ EĞİTİM İLE GENELLEŞTİRİLEBİLİR YÜZ SAHTECİLİĞİ TESPİTİ

Mustafa Hakan Kara
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Uğur Güdükbay
İkinci Tez Danışmanı: Ayşegül Dündar Boral
Nisan 2025

Derin üretken ağlar ile üretilen sahte yüz görüntülerinin giderek daha gelişmiş hale gelmesiyle dijital medyadaki manipülasyonların tespiti önemli ölçüde güçleşmektedir. Bu durum dijital medya güvenilirliğini korumanın ve görsel dezenformasyonla mücadele etmenin önemini vurgulamaktadır. Hâlihazırda kullanılan tespit modelleri, ağırlıklı olarak belirli bir alana özgü verilerle denetimli eğitime dayanmakta olup daha önce karşılaşılmamış tekniklerle üretilen sahte görüntüler karşısında genellikle daha düşük performans göstermektedir. Bu probleme karşı, çeşitli sahtecilik yöntemlerine uyum yeteneğine sahip bir Siyam ağı mimarisiyle üçlü öğrenmeyi kullanan bir yüz sahteciliği tespit yaklaşımı *Trident'ı* sunuyoruz. *Trident*, gerçek örnekleri manipüle edilmiş olanlardan ayıran hassas özellikleri yakalamak üzere eğitilmiştir. Bu süreçte, nüans farklarını izole ederken diğer değişkenleri korumaya yönelik oluşturulmuş üçlüler kullanılmaktadır. Aynı zamanda, yöntemimizin genelleme yeteneğini daha da artırmak üzere çekişmeli olarak eğittiğimiz Sahtecilik Ayırt Edicisini sunmaktayız. Bu rekabetçi bileşen, özellik çıkaran modelimizi belirli bir sahtecilik yönteminden bağımsız olarak, daha önce görülmemiş manipülasyonlara daha dirençli hale getirmektedir. Buna ek olarak sınıflandırıcı bileşene gradyan akışını önleyerek belirli sahtecilik türlerine özgü olan aşırı uyumlanmayı önlüyoruz. Çeşitli test veri setleri üzerinde yapılan kapsamlı değerlendirmeler ve ayrıştırma çalışmaları, yöntemimizin etkinliğini ortaya koymaktadır.

Anahtar sözcükler: Derin sahte tespiti, metrik öğrenme, üçlü öğrenme, alan-çekişmeli eğitim, yüz sahteciliği tespiti.

Acknowledgement

I thank my advisors, Professors Uğur Güdükbay and Ayşegül Dündar Boral, for their continued support and guidance throughout my research. I extend my heartfelt gratitude to my parents, whose unwavering encouragement motivated me through the most demanding phases of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advances in deep-generative networks, particularly Generative Adversarial Networks (GANs) [1] and, more recently, diffusion models [2], have led to increasingly sophisticated visual forgeries, posing significant challenges to digital media integrity and public discourse. As manipulated images become prevalent, traditional supervised approaches have shown limitations in:

- generalization across diverse manipulation techniques and
- detection of previously unseen forgery methods [3].

To address these challenges, we present *Trident* (**Tri**plet learning-based **De**epfake detection **Ne**t**t**work), a triplet learning-based approach designed to generalize across a wide range of forgery techniques, including those not encountered during training. Our method builds upon triplet loss techniques originally introduced for face recognition tasks [4] but adapts them to the more complex objective of detecting forged imagery. By carefully curating triplets that preserve the person ID and scene context, *Trident* encourages the model to isolate subtle, forgery-specific features rather than relying on incidental factors like background or identity. This design ensures that the learned embeddings capture the intrinsic differences between genuine and manipulated samples, thereby improving

generalization and the detection process's robustness.

Trident also leverages domain-adversarial training [5] with a forgery discriminator, which attempts to classify the forgery category given an embedding. This adversarial interplay pushes the embedding generator to produce representations agnostic to specific forgery types, focusing instead on the fundamental artifacts that distinguish real from fake. Such an adversarial mechanism reduces overfitting to known manipulation methods, resulting in discriminative embeddings even when confronted with novel, unseen forgeries.

In addition, we detach siamese network embeddings before passing them to the classifier head, ensuring the classifier's gradients do not influence the embedding network during backpropagation. This maintains the integrity of the embedding space learned through triplet loss, allowing it to focus more on capturing generalizable forgery artifacts.

Our main contributions are

- a triplet learning formulation that isolates forgery-specific features by controlling for identity and scene, ensuring that the learned embeddings emphasize the intrinsic attributes of manipulated content,
- the introduction of an adversarially trained forgery discriminator, guiding the embeddings toward forgery-agnostic representations that remain effective against unseen manipulations, and
- detaching siamese network embeddings before classification to isolate gradient flow.

Extensive experiments demonstrate that the *Trident* framework achieves competitive performance on challenging benchmarks while exhibiting more adaptability than traditional supervised approaches. Our findings highlight the potential of combining triplet learning and domain-adversarial training to detect the evolving landscape of face forgeries.

Chapter 2

Background

2.1 Manifold Assumption

Modern deep learning techniques often hinge on the *manifold assumption*, which posits that high-dimensional data lie on or near a manifold of significantly lower intrinsic dimensionality. In other words, while data such as images or speech signals may be represented as high-dimensional vectors (e.g., pixel intensities for images or spectrogram coefficients for speech), they typically exhibit rich internal correlations that reduce the effective degrees of freedom. This premise allows learning algorithms to focus on a lower-dimensional *intrinsic* space, thus making tasks like classification, regression, or representation learning more tractable.

Figure 2.1 illustrates this principle, highlighting how data scattered in a high-dimensional ambient space can, in reality, be confined to a smooth, curved manifold of reduced dimensionality. This insight motivates a variety of *metric learning* methods (see Section 2.2), whose goal is to map data into a latent space where Euclidean distances (or other standard distance metrics) better capture the underlying semantic relationships among data points.

By leveraging this manifold-based perspective, deep learning methods can learn

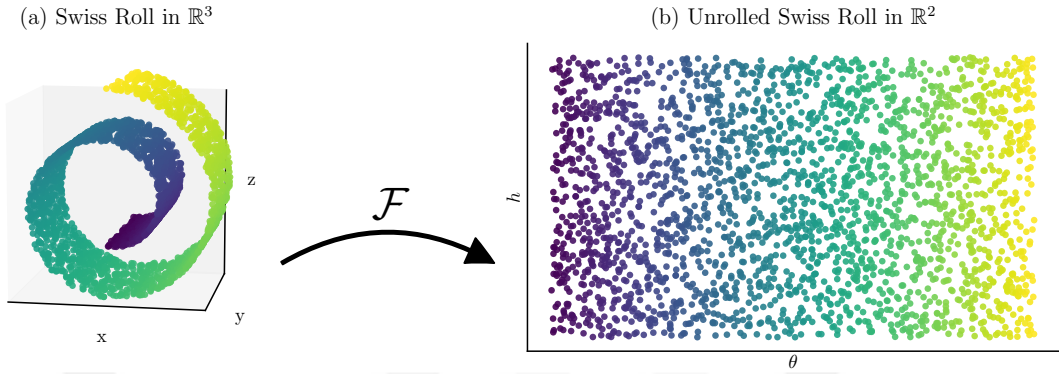


Figure 2.1: A conceptual illustration of the manifold assumption. Although the data points reside in a high-dimensional ambient space, they are well-approximated by (or lie on) a lower-dimensional manifold $\mathcal{M}$. The mapping $\mathcal{F} : \mathcal{X} \to \mathcal{M}$ highlights how the intrinsic dimension of the data can be smaller than the ambient dimension.

hierarchical representations that isolate essential factors of variation. As information propagates through multiple layers of a neural network, high-dimensional data are mapped onto progressively lower-dimensional subspaces, or manifolds, which are more amenable to classification, clustering, or retrieval tasks. This conceptual foundation paves the way for robust *metric learning* algorithms, including triplet-based methods and Siamese networks, to learn feature embeddings aligned with the manifold assumption.

2.2 Metric Learning

Metric learning refers to a broad class of methods aimed at learning an embedding space in which a chosen distance metric (e.g., Euclidean distance) accurately reflects the similarity or dissimilarity between data points. The central objective is to transform the raw input data into a latent space where semantically similar samples are mapped closely while dissimilar ones are placed farther apart.

Traditional machine learning methods may rely on predefined distance metrics (e.g., raw pixel values in image tasks). However, these metrics often fail to capture real-world data's complex structure and semantic relationships. Metric

learning methods instead *learn* the distance function by leveraging labeled data or other supervisory signals, producing feature embeddings that are both robust and meaningful for tasks such as classification, retrieval, or clustering.

A typical metric learning pipeline involves:

1. **Feature extraction:** A model (often a deep neural network) computes high-level representations of the input data.
2. **Loss function design:** A specialized loss function (e.g., contrastive loss or triplet loss) encourages pairs or triplets of samples with certain similarity relationships to be embedded at appropriate distances.
3. **Embedding space construction:** The learned parameters of the model produce an embedding space where the distance metric is better aligned with the task-specific notion of similarity.

In the context of *forgery detection*, metric learning facilitates distinguishing between real and manipulated content based on carefully designed feature representations. The learned embeddings can support classification (deciding whether a sample is real or fake) and retrieval-like processes (e.g., locating the most similar known examples in a database).

In the following subsections, we discuss two widely used metric learning strategies: *contrastive learning* (Section 2.2.1) and *triplet learning* (Section 2.2.3). Both approaches encourage the model to map similar samples to nearby points in the latent space while separating dissimilar samples. However, they differ in how they form training pairs or triplets and the specific objectives they optimize.

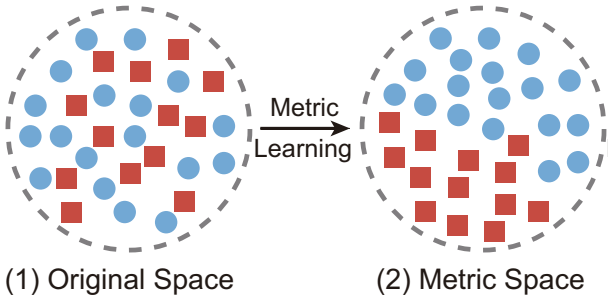


Figure 2.2: An illustration of metric learning. In the original space (left), data points (blue circles and red squares) are interspersed in a way that may not reflect their true semantic relationships. After metric learning (right), semantically similar points are mapped closer together, while dissimilar points are pushed farther apart, yielding a more meaningful embedding space. Adapted from Li *et al.*, "A multilayer framework for online metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 6701–6713, 2022. © IEEE.

2.2.1 Contrastive Learning

Contrastive Learning is a paradigm wherein models learn to distinguish between similar and dissimilar data points via an informative embedding space. Introduced for face verification [6], it operates on *paired* inputs labeled as either "similar" or "dissimilar." The *contrastive loss* function enforces:

- *Minimizing* the distance between embeddings of similar pairs.
- *Maximizing* the distance between embeddings of dissimilar pairs.

Formally, if $d(\cdot, \cdot)$ is a distance metric in the learned space, the loss penalizes large distances $d(x^a, x^p)$ for positive pairs and small distances $d(x^a, x^n)$ for negative pairs. The model learns to cluster semantically related points and separate dissimilar ones by repeatedly processing labeled pairs.

2.2.2 Siamese Networks

A *Siamese Network* comprises multiple identical neural network branches with shared parameters, originally proposed for signature verification [7]. Given multiple input samples, each branch independently produces an embedding, and a distance metric (e.g., Euclidean distance) measures how similar or dissimilar these embeddings are. The shared weights ensure that input samples undergo an identical transformation, as illustrated in Figure 2.3.

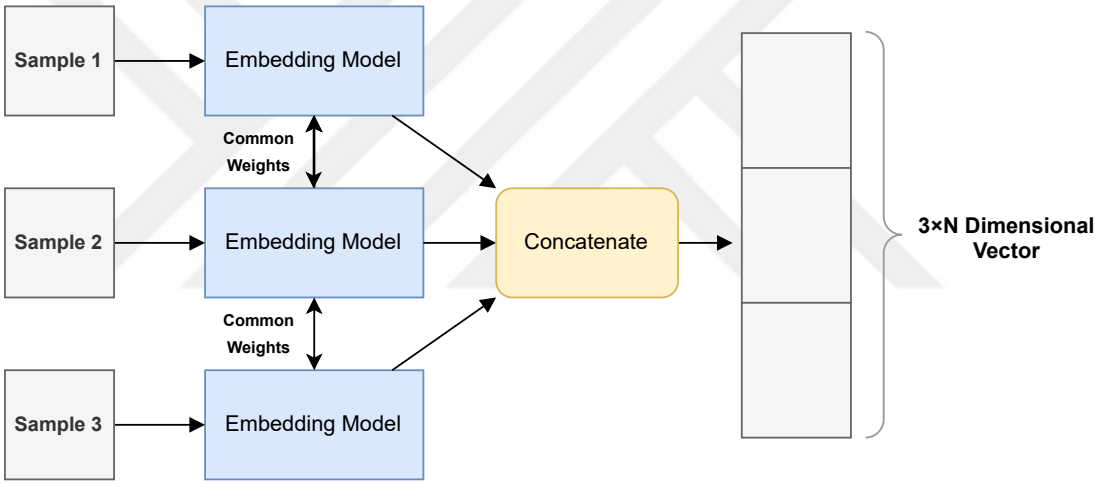


Figure 2.3: A Siamese network architecture. The figure illustrates the core concept behind Siamese networks: an identical embedding function applied to each input sample.

During training, a *contrastive loss* objective is typically employed to draw embeddings of similar pairs closer together and push those of dissimilar pairs apart. Once trained, the network projects inputs into a latent space where canonical distance metrics quantify their semantic relationships meaningfully.

2.2.3 Triplet Learning

Triplet Learning generalizes pairwise comparisons by introducing an anchor-positive-negative sample trio. Popularized by Schroff et al. [4], it employs a *triplet loss* that draws the *anchor* x^a closer to its *positive* x^p (similar to the anchor) while pushing it away from its *negative* x^n (dissimilar to the anchor). A margin m enforces

$$d\big(f(x^a),\, f(x^p)\big) + m \;\leq\; d\big(f(x^a),\, f(x^n)\big),$$

where $f(\cdot)$ denotes the learned embedding function. By simultaneously considering three samples, triplet learning captures more nuanced relationships and encourages the model to discern fine-grained distinctions in the embedding space.

2.3 Adversarial Training

Adversarial training is based on a minimax game from game theory, in which two entities—often the *generator* and *discriminator*—engage in a zero-sum contest. Goodfellow et al. [1] formalized this approach via Generative Adversarial Networks (GANs), proposing the objective:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (2.1)$$

Here, the *discriminator* D aims to accurately classify real vs. generated data, thus *minimizing* its own loss. Concurrently, the *generator* G seeks to *maximize* the discriminator's loss by synthesizing increasingly realistic samples. This interplay forms a zero-sum game, motivating G to refine its generative strategies while D becomes more discriminative.

Adversarial principles extend beyond generative modeling to tasks like representation learning and domain adaptation, where they promote *invariance* to

specific attributes. A notable tool in these contexts is the *Gradient Reversal Layer (GRL)*, which facilitates adversarial alignment of learned features.

One practical and influential architecture that implements this principle was proposed by Ganin et al. [5] and is illustrated in Figure 2.4. This architecture integrates a deep feature extractor, a label predictor optimized for the classification task on the source domain, and a domain classifier designed to differentiate between source and target domains.

A critical component is the adversarial mechanism, introduced by gradient reversal during backpropagation, which ensures that the learned feature representations become invariant to domain-specific differences.

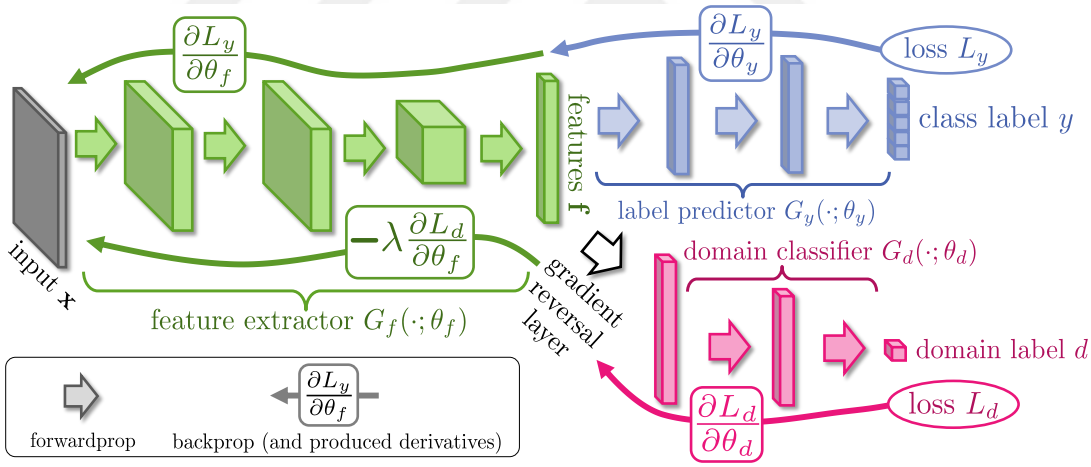


Figure 2.4: A feature extractor generates representations (green) connected to two distinct classifiers: a label predictor optimized on source labels, and a domain classifier aiming to distinguish between source and target domains. The gradient reversal operation ensures adversarial optimization, enforcing the learning of domain-invariant features. Adapted from Ganin et al. [5].

2.3.1 Gradient Reversal Layer

The *Gradient Reversal Layer (GRL)* [5] enables adversarial objectives to be embedded within a standard backpropagation routine. The GRL acts as an identity function in the forward pass, transmitting activations unchanged. During the

backward pass, however, it multiplies the gradient by $-\lambda$ (for a hyperparameter $\lambda > 0$), effectively *reversing* the gradient's direction:

$$G(x) = x, \quad \frac{\partial G}{\partial x} = -\lambda I. \tag{2.2}$$

The GRL induces a *minimax* interplay between different network branches by inverting the gradient flow. For instance, one module may attempt to preserve discriminative signals for classification, while an adversarially trained discriminator tries to exploit those signals. The GRL ensures the embedding module is penalized for retaining easily discriminative cues, thereby fostering *invariant* features.

Chapter 3

Related Work

3.1 Facial Forgery Detection

The proliferation of deep learning techniques has led to sophisticated facial forgery methods, posing significant challenges for visual content authentication [8]. Detecting such forgeries has become a critical area of research in computer vision. Existing facial forgery detection methods can be broadly categorized into spatial, temporal, and frequency domain methods.

3.1.1 Spatial Domain Methods

Spatial domain methods focus on detecting inconsistencies and artifacts within the pixel space of images. Naive detectors use convolutional neural networks (CNNs) to directly classify images as real or fake. Notably, Xception [9] and EfficientNet [10] have been widely adopted as baseline models in deepfake detection tasks due to their strong performance [11].

Advanced spatial methods aim to capture specific artifacts introduced by forgery techniques. Li *et al.* [12] proposed *Face X-Ray*, focusing on detecting the

blending artifacts inherent in face-swapping operations. Chai *et al.* [13] examined the inherent differences between the images captured by cameras and those manipulated by algorithms, improving detection robustness. Zhao *et al.* [14] introduced a multi-attentional framework to emphasize texture details, leveraging a texture enhancement module and attention mechanisms. Nirkin *et al.* [15] detected swapped faces by analyzing discrepancies between facial regions and their surrounding context. Recent works like SBIs [16] used self-blended images to encourage models to learn generalized forgery traces, enhancing generalization capabilities. RECCE [17] employed reconstruction learning on real samples to capture common representations of authentic facial images.

Nguyen *et al.* [18] explored using Capsule Networks to detect fake images and videos. Capsule Networks effectively capture hierarchical spatial relationships between features, aiding in forgery detection by modeling spatial hierarchies, although they face challenges with generalizing to unseen data. In addition, Nguyen *et al.* [19] proposed a multi-task learning framework that jointly detects manipulated facial images/videos and segments the corresponding forged regions. Their approach uses a Y-shaped decoder to generate both a binary detection output and a segmentation mask, and it further integrates a reconstruction branch to improve generalizability.

3.1.2 Temporal Domain Methods

Temporal domain methods exploit inconsistencies across video frames to detect forgeries. Yang *et al.* [20] focused on detecting inconsistent head poses caused by independent frame manipulation. Zheng *et al.* [21] observed that frame-by-frame face forgery generation leads to noticeable flickering and discontinuity, which can be exploited for detection. LipForensics [22] concentrated on temporal inconsistencies in mouth movements, providing robustness against various manipulation techniques. Choi *et al.* [23] proposed exploiting inconsistencies in style latent vectors between frames for effective deepfake detection.

Gu *et al.* [24] introduced Spatio-Temporal Inconsistency Learning (STIL) for

video deepfake detection. This method combines Spatial Inconsistency Modules (SIM) and Temporal Inconsistency Modules (TIM) to capture spatial and temporal mismatches, achieving superior detection accuracy by fusing spatial and temporal information.

Amerini *et al.* [25] employed optical flow-based CNNs to detect visual inconsistencies in deepfake videos. Focusing on motion cues and discrepancies in temporal coherence, their approach effectively identifies manipulated areas, especially in compressed videos.

3.1.3 Frequency Domain Methods

Frequency domain methods detect forgeries by analyzing anomalies in the frequency spectrum of images. Qian *et al.* [26] introduced F^3 -Net, using a two-branch framework with a cross-attention module to mine frequency-aware clues. Li *et al.* [27] proposed frequency-aware discriminative feature learning, using an adaptive frequency feature generation module to extract differential features from different frequency bands. Liu *et al.* [28] presented Spatial-Phase Shallow Learning (SPSL), focusing on phase discrepancies in the frequency domain for improved detection performance. Frank *et al.* [29] analyzed GAN-based manipulations from a frequency-domain perspective and revealed that upsampling operations in typical GAN architectures leave distinct artifacts. Focusing on these frequency-domain inconsistencies, their method exposes deepfake images created by various neural architectures, showing that GAN-generated content exhibits telling structural patterns in high-frequency bands.

3.1.4 Biological Signal-Based Methods

Biological signal-based methods leverage physiological cues to detect deepfakes. Ciftci *et al.* [30] leveraged biological signals, specifically heartbeat and blood

flow, to detect deepfake videos. This method focuses on residuals in physiological signals that GAN-based forgeries struggle to replicate, providing an effective approach to detect subtle manipulation. Stefanov *et al.* [31] further advanced physiological-based detection by combining visual representations of biological signals with graph convolutional networks. Their method enhances detection accuracy by fusing visual features with physiological signals, offering a robust multi-modal detection framework.

3.1.5 Transformer-Based Methods

Vision Transformers (ViTs) have emerged as a powerful architecture for deepfake detection due to their ability to model local and global image relationships through self-attention mechanisms. As identified in a recent comprehensive survey by Wang et al. [32], ViT-based approaches can be broadly categorized into standalone, sequential, and hybrid architectures.

Among standalone approaches, ISTVT [33] introduced a novel combination of spatial and temporal attention to capture inconsistencies across video frames. Building on this temporal modeling, M2TR [34] proposed simultaneously analyzing images at multiple scales through parallel transformer branches while incorporating frequency domain features to improve robustness against compression artifacts.

In the sequential category, ICT [35] focused on leveraging transformers to verify identity consistency between different facial regions. UIA-ViT [36] made a significant advance by eliminating the need for pixel-level forgery annotations through an unsupervised approach based on feature distribution modeling.

Recent works have explored hybrid approaches combining transformers with other architectures. Khan *et al.* [37] integrated UV texture maps with transformers for improved pose-invariant detection, while GenConViT [38] combined generative models with vision transformers to better capture manipulation artifacts.

3.2 Generalizable Forgery Detection

Despite the progress in facial forgery detection, generalizing to unseen forgery methods remains challenging. Recent work has focused on developing more generalizable approaches to deepfake detection.

Yu *et al.* [39] attributed synthetic images to their source GANs by learning unique fingerprints from different generative models. Dang *et al.* [40] employed attention mechanisms to refine feature representations, improving the detection of manipulated faces across different forgery techniques. Ni *et al.* [41] proposed *CORE*, a simple yet effective framework that explicitly enforces representation consistency across augmented versions of the same image. By regularizing the cosine distances between different augmentations. Zhao *et al.* [42] proposed learning self-consistency to enhance the generalization of detectors. Huang *et al.* [43] addressed implicit identity leakage, proposing a framework to improve generalization by disentangling identity information. Zhai *et al.* [44] introduced a weakly supervised detection approach, requiring only binary image-level labels for training. Yan *et al.* [45] addressed the challenge of generalizing to unseen forgery types by decomposing images into forgery-irrelevant, method-specific, and common forgery features. Through a multi-task learning strategy that predicts both the forgery category and a binary real/fake label, their approach explicitly disentangles method-specific artifacts, extracting truly shared forgery features.

Ojha *et al.* [46] proposed using a pretrained encoder followed by traditional machine learning classifiers to avoid domain specificity often seen in end-to-end deep learning models. Reiss *et al.* [47] introduced an approach mimicking fact-checking processes, using off-the-shelf encoders to detect previously unseen deepfake attacks. Tan *et al.* [48] presented the LGrad framework, which leverages gradients as generalized representations to detect artifacts in GAN-generated images, including those generated by unseen models.

Contrastive and metric learning methods have shown promise in enhancing the generalizability of deepfake detection by learning discriminative features. Fung

et al. [49] proposed DeepfakeUCL, an unsupervised approach applying image transformations to distinguish real from fake samples. Sun *et al.* [50] introduced *Contrastive Pseudo Learning* for open-world deepfake attribution, incorporating global-local voting and confidence-based soft pseudo-labeling. Xu *et al.* [51] developed a supervised contrastive learning method for generalizable and explainable detection, fusing their model with Xception for improved performance on unknown attacks. Hong *et al.* [52] proposed a unified approach for classification and localization using multiple instance learning and multi-label ranking.

For video detection, Gu *et al.* [53] introduced Hierarchical Contrastive Inconsistency Learning, capturing temporal inconsistencies from local and global perspectives. Larue *et al.* [54] presented SeeABLE, using soft discrepancies and bounded contrastive learning to improve generalization to unknown deepfake types. Liang *et al.* [55] proposed a depth map-guided triplet network, using depth prediction to reflect differences between real and fake faces and learn discriminative features. Kumar *et al.* [56] introduced a metric learning approach using triplet networks, effective in high-compression scenarios.

Mittal *et al.* [57] presented an audio-visual deepfake detection method using affective cues, employing a network inspired by Siamese architecture and triplet loss. Beuve *et al.* [58] introduced methods to infer a binary class from a multi-class backbone, leveraging dummy triplet loss. Jain *et al.* [59] demonstrated that training for attribution with triplet loss improves generalization in cross-database scenarios.

3.3 Adversarial Training

Minimax-style adversarial training in deep learning was introduced through Generative Adversarial Networks (GANs), which cast the training process as a competition between a generator and a discriminator. The generator aims to synthesize data that resembles real samples, while the discriminator attempts to distinguish between real and generated samples. Through this adversarial process, both

networks iteratively improve—the generator produces increasingly realistic data while the discriminator becomes better at detection until reaching an equilibrium.

In supervised or semi-supervised learning, adversarial training has evolved to improve feature transferability and invariance. For example, Ganin and Lempitsky [5] introduced the Gradient Reversal Layer (GRL) to learn domain-invariant representations for unsupervised domain adaptation, effectively reducing feature distribution discrepancies across different domains. Similar adversarial ideas were applied in EANN [60], where the network learned to produce event-invariant embeddings to generalize across different events in fake news detection. This work demonstrates the strength of adversarial objectives in promoting general-purpose, robust feature extraction—an insight that inspires our approach to face forgery detection.

Chapter 4

Method

Unlike previous approaches, we apply triplet learning to deepfake detection in a controlled setting. By selecting triplets that maintain consistent person ID and scene, we isolate forgery-specific features, prompting the model to focus on subtle and discriminative artifacts introduced by the forgery process rather than incidental variations in identity or background.

We integrate a domain-adversarial training component to drive the network towards learning forgery-agnostic features, employing a forgery discriminator to classify specific manipulation methods. On the other hand, the embedding generator seeks to produce representations indistinguishable to this discriminator, thus pushing the model to rely on universal markers of manipulation rather than forgery-type-specific artifacts. This adversarial interplay leads to more generalized embeddings that better handle emerging and previously unseen forgery techniques.

These components collectively aim to improve the generalizability and interpretability of deepfake detection—overcoming the limitations of current supervised methods and offering robust detection even when faced with novel or evolving forgery strategies.

4.1 Triplet Learning Framework

Trident employs a triplet learning setting, building upon the foundations of contrastive learning and Siamese networks. Contrastive learning, originally proposed for face verification tasks [6], aims to distinguish samples by teaching the model to pull similar items closer and push dissimilar ones apart in the feature space. It employs contrastive loss on pairs of data points, penalizing large distances between similar and small distances between dissimilar pairs.

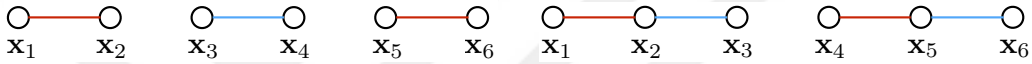


Figure 4.1: Contrastive vs. triplet embedding. Adapted from H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pp. 4004–4012, 2016. © IEEE.

Siamese networks, introduced by [7], involve feeding pairs of samples into identical neural networks with shared weights. This architecture typically measures the similarity of two inputs regarding learned feature representations. As described in [4], triplet learning extends these concepts by introducing a third element, creating anchor-positive-negative sample triplets.

This approach aims to bring the anchor and positive samples closer together while pushing the anchor and negative samples farther apart in the feature space. We leverage this triplet learning setting in *Trident*, training a projection head with triplets consisting of two real samples (anchor and positive) and one fake sample (negative).

Our approach leverages this idea to learn the fine-grained features that distinguish genuine samples from forgeries while controlling for other variables such as scene and person ID. This allows us to capture nuances and forces the model to discern subtle differences and similarities, potentially improving our model's ability to generalize to previously unseen forgeries.

4.2 Model Architecture

Our model architecture, shown in Figure 4.6, involves a Siamese network structure with a triplet learning setup to distinguish real samples from fake ones. By curating triplets with consistent person ID and scene, we apply triplet learning to deepfake detection in a controlled setting. This controlled triplet learning setting isolates forgery-specific features, allowing the model to focus on finer details and more discriminative features related to the forgery process rather than incidental variations in identity or background. This section provides an in-depth explanation of each component in our architecture.

4.2.1 Siamese Network and Embedding Generation

We process each triplet of inputs through three parallel Siamese networks with shared weights, generating embeddings that capture fine-grained, distinguishing features between genuine and forged samples. Our triplet loss structure encourages the model to minimize the distance between the anchor and positive samples while maximizing the distance between the anchor and negative samples in feature space. By designing the embeddings, we enable the model to learn representations robust to variations such as person ID or scene context, improving its generalization ability.

4.2.2 Depthwise-Separable Convolution

The CNN-based backbones we employed are built upon depthwise-separable convolutions—a design choice inspired by architectures such as Xception [9] and EfficientNet [10]. These models replace standard convolutional layers with a factorized operation that first applies a depthwise convolution (processing each input channel separately) followed by a pointwise 1×1 convolution (combining the per-channel features). This decomposition reduces the number of parameters

and computations while retaining—and in some cases enhancing—the representational power. Empirical studies have shown that such architectures perform well in forgery detection tasks by effectively capturing fine-grained artifacts, making them popular baselines in recent works.

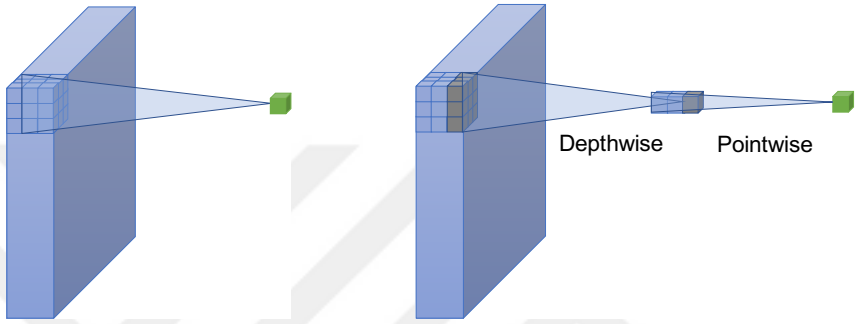


Figure 4.2: Standard convolution (left) and depthwise-separable convolution (right). In standard convolution, a filter is applied across all input channels simultaneously, resulting in high computational cost and many parameters. In contrast, depthwise-separable convolution decomposes this operation into two steps—a depthwise convolution that processes each channel independently followed by a pointwise 1×1 convolution aggregating the outputs, thereby reducing the parameter count and computational load. Adapted from Y. Guo, Y. Li, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 8368–8375, 2019. © IEEE.

4.2.3 Transformer Architecture

Transformer architecture [61] represents a significant advancement in deep learning, introducing attention mechanisms that establish direct relationships between all elements in an input sequence. At the core of transformers is the self-attention mechanism, which computes dynamic weights for each element in a sequence based on its relevance to all other elements:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, input representations are projected into query (Q), key (K), and value

(V) spaces. The scaled dot-product between queries and keys determines attention weights, which are applied to values to produce context-aware representations.

Transformers typically employ multi-head attention, allowing parallel attention processes with different learned projections:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

This multi-head approach enables the model to jointly attend to information from different representational subspaces, capturing diverse relationship patterns within the data. The transformer block combines this attention mechanism with feed-forward networks, layer normalization, and residual connections to form a powerful processing unit that has transformed numerous domains in artificial intelligence.

4.2.4 Vision Transformer

Vision Transformers (ViT) [62] adapt the transformer architecture to computer vision tasks. Unlike convolutional neural networks that process images through hierarchical local filters, ViT treats images as sequences of patches, applying transformer blocks to capture global dependencies directly.

The ViT processing pipeline transforms images into a format suitable for transformer processing. Initially, input images are divided into non-overlapping patches, typically of size 16×16 pixels. These patches are then linearly projected to create token embeddings of consistent dimensionality. To retain spatial information that would otherwise be lost in this tokenization process, position embeddings are added to these patch tokens. The resulting sequence is processed through a series of transformer encoder blocks, each applying self-attention mechanisms and feed-forward networks to capture relationships between all patches regardless of their spatial distance.

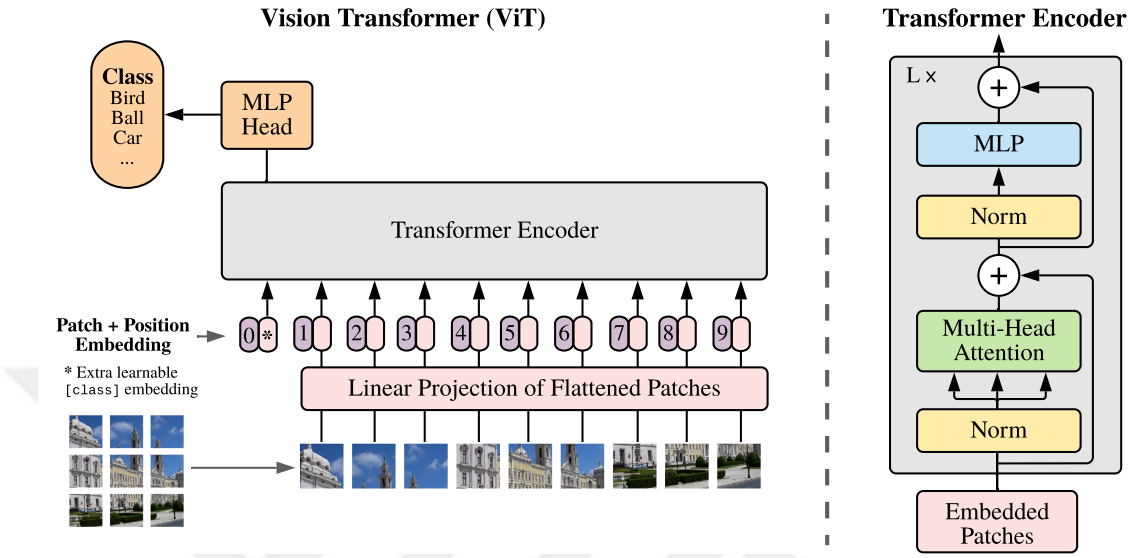


Figure 4.3: Architecture of the Vision Transformer (ViT). The image is split into fixed-size patches, which are linearly embedded along with position embeddings. A standard transformer encoder processes the resulting sequence of vectors. The output embedding of a special [class] token is used for classification tasks. The transformer encoder consists of alternating layers of multi-headed self-attention and MLP blocks, with layer normalization and residual connections. Adapted from A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," Proceedings of the International Conference on Learning Representations, 2021.

This patch-based approach breaks away from the locality bias inherent in convolutional operations, allowing direct modeling of relationships between distant image regions. The self-attention mechanism provides adaptive receptive fields that dynamically adjust to focus on the most informative regions for the task at hand.

In facial analysis, this property enables the model to capture relationships between features such as eyes, mouth, and skin texture without being constrained by fixed convolutional kernels. The global context modeling capacity of transformers makes them particularly suitable for tasks requiring holistic understanding of facial structure and consistency.

4.2.5 Masked Autoencoder

Masked Autoencoders (MAE) [63] represent a self-supervised learning approach that trains models to reconstruct images from partially masked inputs. Drawing inspiration from masked language modeling in NLP, MAEs randomly mask a high proportion (typically 75%) of image patches and task the model with reconstructing the missing content.

The MAE framework operates through a multi-stage process designed to learn meaningful representations without explicit supervision. The approach begins by randomly masking a substantial portion of image patches from the input. Unlike standard autoencoders, MAEs process only the visible (unmasked) patches through an encoder, significantly reducing computational requirements. The encoder outputs are then combined with learnable mask tokens that serve as placeholders for the missing regions. Finally, a decoder processes this combined representation to reconstruct the original image, focusing particularly on accurately recovering the content of the masked regions.

This approach offers several advantages for representation learning. By masking a large portion of the input, the model must learn meaningful semantic and structural features rather than memorizing pixel-level details. The asymmetric encoder-decoder design maintains computational efficiency despite the high masking ratio. The reconstruction objective encourages the model to develop holistic representations that capture underlying image structure and semantics.

Notably, Masked Autoencoders (MAE) employ the same self-supervised learning paradigm used in image inpainting systems and large language model (LLM) pretraining—predicting masked or missing content from contextual information. This fundamental masking-and-reconstruction approach has demonstrated remarkable effectiveness across multiple domains, enabling models to learn rich, transferable representations without reliance on manually annotated data.

The MAE training objective is typically formulated as:

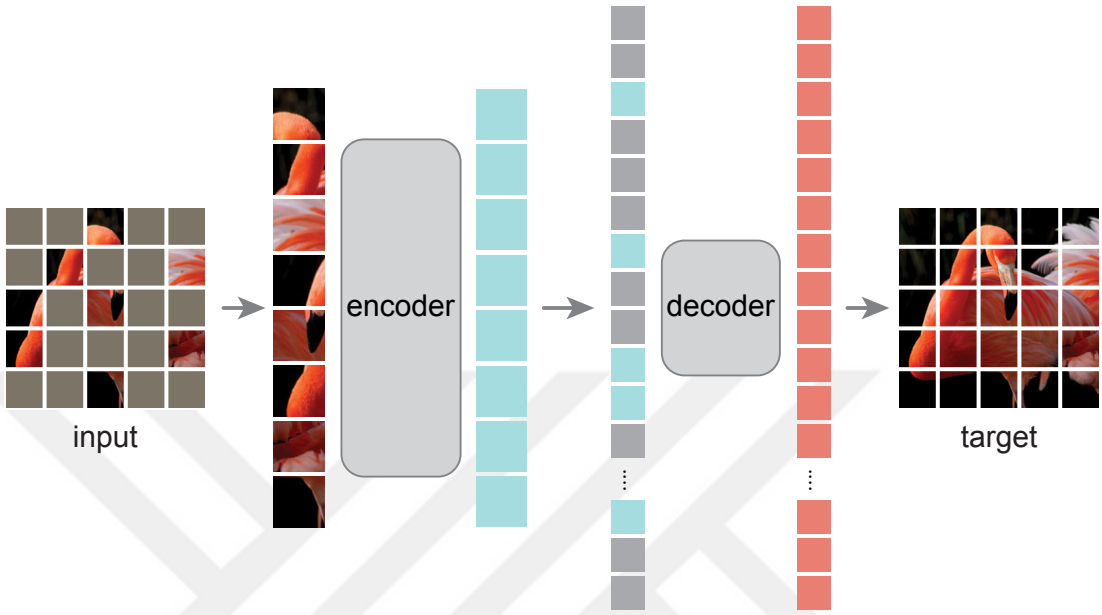


Figure 4.4: Masked Autoencoder (MAE) architecture. The approach randomly masks a significant portion of the input image patches (shown in gray). Only the visible patches are processed by the encoder, making the approach computationally efficient. The encoded visible patches, combined with mask tokens, are then passed to the decoder which reconstructs the original image. This self-supervised approach forces the model to learn meaningful image representations without requiring labeled data. Adapted from K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR '22, pp. 16000–16009, June 2022. © IEEE.

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|M|} \sum_{i \in M} \|x_i - \hat{x}_i\|^2$$

where M is the set of masked patch indices, x_i is the original content of patch i, and $\hat{x}_i$ is the reconstructed content.

4.2.6 MARLIN

MARLIN (Masked Autoencoder for facial video Representation LearnINg) [64] extends the Vision Transformer and Masked Autoencoder frameworks to facial

video analysis. Its key innovation is a structured masking strategy targeting semantically significant facial regions—eyes, nose, mouth, lips, and skin areas.

This approach creates a challenging reconstruction task that drives comprehensive facial understanding. By reconstructing these information-rich regions, MARLIN learns representations capturing both fine-grained details and broader structural relationships. This reconstruction of masked spatiotemporal facial information functions as an effective self-supervised pretext task.

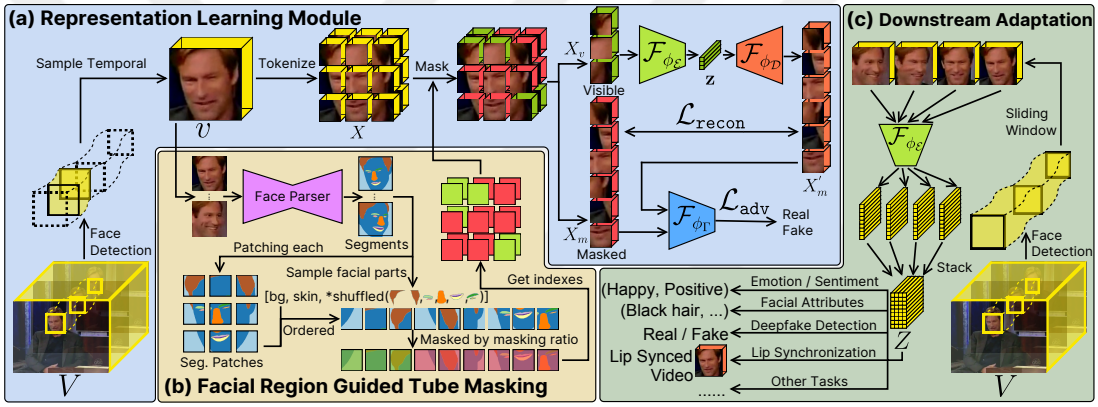


Figure 4.5: MARLIN architecture overview. (a) The representation learning module samples facial video frames and processes them through a transformer-based encoder-decoder. (b) Unlike standard MAE approaches, MARLIN employs facial region guided tube masking that specifically targets semantically meaningful facial regions such as eyes, mouth, and skin. (c) The pre-trained MARLIN encoder. Adapted from Z. Cai, S. Ghosh, K. Stefanov, A. Dhall, J. Cai, H. Rezatofighi, R. Haffari, and M. Hayat, "Marlin: Masked autoencoder for facial video representation learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR '23, pp. 1493–1504, 2023. © IEEE.

MARLIN uses non-annotated facial videos from web collections to learn robust facial embeddings. This self-supervised methodology has shown strong transferability across multiple domains including Facial Attribute Recognition, Facial Expression Recognition, Deepfake Detection, and Lip Synchronization.

In our research, we employ the MARLIN encoder for the transformer-based variant of our embedding model. We selected it for its extensive pretraining on diverse facial data and demonstrated ability to extract nuanced facial features

that generalize across domains. These learned representations enhance our system's sensitivity to the subtle inconsistencies and artifacts present in manipulated facial content.

4.2.7 Domain-adversarial Training

The primary objective of our framework is to encourage the Siamese network to generate *forgery-agnostic embeddings*—representations that can effectively distinguish real from fake content without relying on forgery-specific features. We introduce a forgery discriminator to support this goal, which is adversarial in guiding the embedding generator toward more generalized, forgery-invariant representations.

The forgery discriminator is trained to classify the forgery category for embeddings from multi-forgery datasets such as FaceForensics++ [11]. By attempting to identify the forgery type, the discriminator indirectly pushes the embedding generator to avoid encoding forgery-specific artifacts. This results in more generalizable embeddings that capture the underlying characteristics of fake content.

The forgery discriminator consists of multiple fully connected layers, with an output layer producing logits for each forgery category. This adversarial setup encourages the embedding generator to focus on broad real-vs-fake distinctions rather than overfitting to details specific to particular forgery techniques, thereby making the model more robust to unseen manipulations.

This adversarial interaction between the forgery discriminator and the embedding generator is key to generating robust, forgery-agnostic embeddings that enhance performance across diverse forgery types.

4.2.8 Gradient Reversal Layer

The Gradient Reversal Layer (GRL) is defined such that it acts as an identity mapping in the forward pass but reverses and scales the gradient by a factor λ in the backward pass. Let x be the input embeddings and $\lambda > 0$:

$$\text{Forward: } G(x) = x \tag{4.1}$$

During the backward pass, the gradient is multiplied by $-\lambda$:

$$\text{Backward: } \frac{\partial G}{\partial x} = -\lambda I \tag{4.2}$$

where I is the identity matrix. This scaling factor λ allows fine-grained control over the adversarial signal.

The GRL is placed between the embedding generator and the forgery discriminator. During forward propagation, it acts as a pass-through layer, but during backpropagation, it reverses the gradient. This reversal encourages the embedding generator to produce feature representations that confuse the forgery discriminator, thereby learning more generalizable, forgery-invariant embeddings. In other words, adversarial training with the GRL aligns the embedding generator's objective with producing domain-invariant features, preventing the model from overfitting to known forgery types.

4.2.9 Classifier Head Detachment

We isolate the Forgery Detector (real-fake classification head) from the embedding generation process to further enhance generalization. In practice, we block backpropagation from the *Forgery Detector* into the shared embedding layers. By preventing gradient flow from the binary classification task back into the embedding space, we ensure that the embeddings remain focused on forgery-related

features rather than overfitting to the particularities of the classification head's decision boundary. This separation helps maintain a more stable and general-purpose embedding space, improving the model's adaptability to previously unseen forgeries.

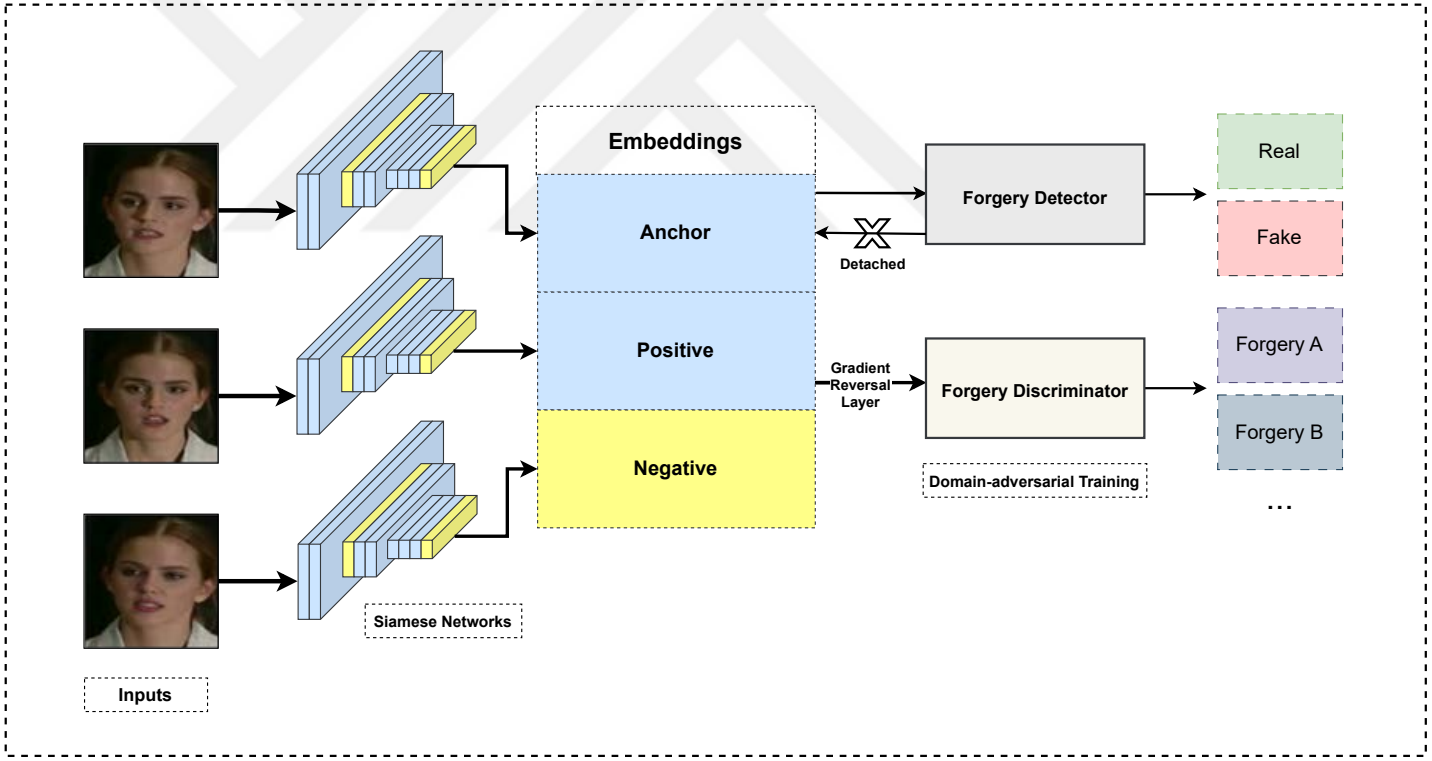


Figure 4.6: Overview of *Trident's* high-level architecture. *Trident* employs a Siamese network with triplet learning to generate discriminative embeddings from input face triplets. The network processes triplets of anchor, positive, and negative samples through shared weights, encouraging similar embeddings for real samples while distancing fake samples. The forgery detector branch performs binary real-fake classification, while the parallel forgery discriminator branch classifies the specific forgery type through multiple categories. We integrate a Gradient Reversal Layer (GRL) between the embedding generator and forgery discriminator, creating a domain-adversarial training setup that drives the model to learn forgery-agnostic features.

4.2.10 Preprocessing

Before constructing the triplets, we process each video by detecting faces employing FaceXZoo [65]. To ensure consistency across the dataset, we extract 32 frames per video and use the detected facial regions to avoid irrelevant parts or artifacts.

4.3 Objective Functions

The training objective in *Trident* is optimized using a hybrid loss function that combines *Binary Cross-Entropy (BCE) loss*, *Triplet loss*, and an *Adversarial loss*. This composite loss is designed to enhance class discrimination (via BCE), improve feature embedding separation (via Triplet loss), and promote forgery-agnostic embedding generation (via Adversarial loss).

The total loss $\mathcal{L}_{\text{total}}$ is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}} + \alpha \cdot \mathcal{L}_{\text{triplet}} + \beta \cdot \mathcal{L}_{\text{adv}} \tag{4.3}$$

where α and β are weighting factors that balance the contributions of the individual loss components.

4.3.1 Binary Cross-Entropy Loss (BCE)

The Binary Cross-Entropy loss $\mathcal{L}_{\text{BCE}}$ is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right) \tag{4.4}$$

where N is the number of samples, y_i is the ground truth label, and $\hat{y}_i$ is the

predicted probability of the sample being a forgery. This loss encourages accurate classification by minimizing the error between the predicted and actual labels.

4.3.2 Triplet Loss

The Triplet loss $\mathcal{L}_{\text{triplet}}$ is formulated as:

$$\mathcal{L}_{\text{triplet}} = \frac{1}{N} \sum_{i=1}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 - m \right]_+ \tag{4.5}$$

where x_i^a, x_i^p, and x_i^n represent the anchor, positive, and negative samples, respectively, and m is the margin that enforces a minimum separation between positive and negative pairs. This loss function brings the anchor and positive embeddings closer together while pushing the anchor and negative embeddings farther apart, thereby enhancing feature distinctiveness.

4.3.3 Adversarial Loss

We employ an *Adversarial loss* inspired by the minimax game in Generative Adversarial Networks (GANs) to further promote forgery-agnostic embeddings. In this setup, the embedding generator aims to create embeddings challenging for the forgery discriminator to classify into specific forgery categories, effectively driving the embeddings toward a more generalized, forgery-invariant representation.

The adversarial loss $\mathcal{L}_{\text{adv}}$ is formulated as:

$$\mathcal{L}_{\text{adv}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \hat{p}_{i,j} \log(\hat{p}_{i,j}) \tag{4.6}$$

where $\hat{p}_{i,j}$ is the predicted probability of the i-th embedding belonging to the j-th forgery category, and K is the number of forgery categories. This adversarial

objective maximizes the entropy of the discriminator's predictions, encouraging the generator to produce embeddings that lack forgery-specific cues. The generator learns robust representations across forgery types through this adversarial interplay, enhancing the model's adaptability to unseen manipulations.

4.4 Triplet Formation

We constructed triplets comprising anchor, positive, and negative samples to train our model effectively. This approach encourages the network to learn embeddings where genuine and manipulated images of the same individual are mapped to distinct regions in the feature space. We formed the triplets as follows:

- *Anchor*: An image (real or fake) of a specific individual.
- *Positive*: Same individual, same authenticity as anchor, different timestamp.
- *Negative*: Same individual, same timestamp as anchor, opposite authenticity.

We partitioned the real and fake samples into two halves for each identity. Denote these partitions as R_1, R_2 (real samples) and F_1, F_2 (fake samples). We ensured anchor and negative samples are temporally aligned. That is, for a given timestamp, we used corresponding frames from both real and fake videos to maintain consistent facial expressions and movements, isolating manipulations as the primary differentiating factor.

We then systematically generated all possible triplet configurations for each identity, alternating which partition serves as anchor, positive, and negative samples, as illustrated in Algorithm 1.

This formation ensures that the triplets control for identity, scene, and temporal factors. By alternating between real and fake samples as anchors and positives

Algorithm 1 Triplet Generation

function GENERATETRIPLETS(RealSamples, FakeSamples)
  Divide RealSamples into two parts: R_1 and R_2
  Divide FakeSamples into two parts: F_1 and F_2
  Initialize empty collections for Triplets and Labels
  for each identity **do**
    Add (R_1, R_2, F_1) to Triplets with label $(0, 0, 1)$
    Add (F_1, F_2, R_1) to Triplets with label $(1, 1, 0)$
    Add (R_2, R_1, F_2) to Triplets with label $(0, 0, 1)$
    Add (F_2, F_1, R_2) to Triplets with label $(1, 1, 0)$
  end for
  return Triplets, Labels
end function

and selecting negatives that match identity but differ in authenticity, we focus the model's learning on the subtle artifacts introduced by manipulation techniques rather than identity-specific features.

Chapter 5

Experiments

We evaluated the proposed method across three experimental scenarios using the FaceForensics++ (FF++) dataset [11] and conducted cross-dataset evaluation on Google's Deepfake Detection (DFD) dataset [66]. This section details the experimental setups, implementation specifics, and the configurations employed for the ablation studies.

5.1 Area Under the Curve

To evaluate the performance of our method, we employ the *Area Under the Receiver Operating Characteristic Curve (AUC)*, a widely used metric in Deepfake detection literature. The AUC provides a threshold-independent measure of a classifier's ability to distinguish between real and fake samples.

The AUC can be interpreted as the likelihood that a randomly selected positive sample (fake) receives a higher predicted score than a randomly selected negative sample (real). Formally, the AUC is defined as:

$$\text{AUC} = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \mathbb{I}(s_i > s_j) + \frac{1}{2}\mathbb{I}(s_i = s_j), \tag{5.1}$$

where:

- N_+ and N_- represent the total number of positive (fake) and negative (real) samples, respectively.
- s_i is the predicted score for the i-th positive sample.
- s_j is the predicted score for the j-th negative sample.
- $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the argument is true and 0 otherwise.

The second term in (5.1), $\frac{1}{2}\mathbb{I}(s_i = s_j)$, accounts for ties between predicted scores by assigning them half the weight.

5.1.1 AUC and the ROC Curve

The AUC is computed as the integral of the Receiver Operating Characteristic (ROC) curve, which plots the *True Positive Rate (TPR)* against the *False Positive Rate (FPR)* at various classification thresholds. The TPR and FPR are defined as follows:

$$TPR = \frac{TP}{TP + FN}, \tag{5.2}$$

$$FPR = \frac{FP}{FP + TN}, \tag{5.3}$$

where TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively.

5.1.2 Computation of AUC

The ROC curve is integrated using the trapezoidal rule to compute AUC numerically. Given a set of n thresholds, the AUC is expressed as:

$$\text{AUC}=\sum_{k=1}^{n-1}\frac{(FPR_{k+1}-FPR_k)\cdot(TPR_k+TPR_{k+1})}{2}, \tag{5.4}$$

where FPR_k and TPR_k represent the values of the False Positive Rate and True Positive Rate at the k^{th} threshold.

AUC is beneficial in Deepfake detection due to its robustness to class imbalance and ability to evaluate classifier performance across all possible decision thresholds. This threshold independence ensures a reliable assessment of the model's capability to distinguish between real and fake samples.

5.2 Datasets

To evaluate the generalization capability of our approach, we employed several benchmark datasets with diverse forgery techniques and quality levels:

FaceForensics++ (FF++) [11] contains 1,000 original videos and their manipulated counterparts using four methods: Deepfakes, Face2Face, FaceSwap, and NeuralTextures. Videos are available at multiple compression levels, facilitating evaluation under varying quality conditions.

Celebrities DeepFake (Celeb-DF) [67] includes two versions: **Celeb-DF (v1)** consists of 408 real videos and 795 corresponding DeepFake videos, and **Celeb-DF (v2)** significantly expands upon this, comprising 590 real videos and 5,639 corresponding DeepFake videos with minimal visual artifacts. Version 2 is particularly challenging for detection due to its realistic visual quality, closely matching real-world DeepFake videos circulated online.

Table 5.1: Breakdown of Benchmarks

Dataset	Pristine	Manipulated	Total
Celeb-DF (v1) [67]	408	795	1,203
Celeb-DF (v2) [67]	590	5,639	6,229
FaceForensics++ [11]	1,000	4,000	5,000
DFD [66]	363	3,000	3,363
DFDC [68]	19,154	86,654	105,808
DFDCP [69]	1,131	4,119	5,250
UADFV [20]	49	49	98

DeepFake Detection (DFD) [66] released by Google contains thousands of Deepfake videos featuring consenting actors in controlled environments. It encompasses various facial expressions, head poses, and lighting conditions.

DeepFake Detection Challenge (DFDC) [68] released by Facebook contains over 100,000 videos created using various deepfake techniques. It features diverse subjects across different ages, ethnicities, and lighting conditions. We also use its preview version **DeepFake Detection Challenge Preview (DFDCP)** [69], which contains a smaller subset of videos but maintains similar diversity, making both datasets particularly challenging for generalization.

University at Albany DeepFake Videos (UADFV) [20] includes authentic and DeepFake manipulated videos specifically curated to expose inconsistencies in head poses—a common artifact in early DeepFake synthesis methods.

Table 5.1 provides a concise summary of these datasets, including the number of pristine (original) and manipulated (fake) samples.

5.3 Evaluation Scenarios

We designed three experimental scenarios to evaluate the effectiveness of the proposed method and analyze the contributions of its components:

- **Scenario 1: Forgery-Specific Training**. Models were trained on only

two forgery types (*Deepfakes* and *NeuralTextures*) from the FF++ dataset. The trained models were then tested on all forgery types in FF++ to assess their ability to generalize to unseen manipulation techniques. This scenario examined the impact of the Gradient Reversal Layer (GRL) and Triplet Learning.

- **Scenario 2: Full Dataset Training**. Models were trained and tested on all forgery types in the FF++ dataset. This scenario measured the overall performance of the proposed method when exposed to a diverse set of forgeries during training.
- **Scenario 3: Cross-Dataset Evaluation**. Models trained on the FF++ (HQ) dataset were evaluated on the DFD dataset. This scenario tested the models' generalization to a completely different dataset with variations in data distribution and manipulation techniques.

5.4 Implementation Details

5.4.1 Network Architecture

We employed both EfficientNet-B4 [10] (CNN) and MARLIN [64] (ViT) as backbone architectures for feature extraction to explore complementary representation capabilities.

We also designed a `NetworkTree` structure to implement the proposed architecture, enabling flexible configuration and easy insertion or modification of components. The overall network architecture forms a tree with the root being the embedding model and two child modules:

- **Forgery Detector**: A binary classifier that predicts whether an input image is real or fake, utilizing the embedding from the feature extractor.

- **Forgery Discriminator**: An auxiliary classifier that predicts the forgery type, used with the GRL to encourage the embedding model to learn features invariant to specific forgery techniques.

5.4.2 Training Configuration

Table 5.2: Hyperparameters

Hyperparameter	**Value**
Learning rate	0.0001
Batch size	4
Triplet Loss margin (m)	1.0
Adversarial loss weight (λ)	1.0
Number of epochs	30

We employed the hyperparameters presented in Table 5.2 to achieve optimal validation performance. We used the Adam optimizer [70] with a learning rate of 0.0001. We chose a batch size of 4 to ensure convergence stability. We set the Triplet Loss margin (m) to 1.0 and fixed the Adversarial loss weight (λ) for the Gradient Reversal Layer (GRL) at 1.0. We trained our models for a maximum of 30 epochs. We selected the best model based on the lowest Binary Cross-Entropy (BCE) loss achieved on the validation set.

5.5 Results

This section presents the experimental results of our framework. The performance is evaluated quantitatively through ablation studies combined with intra-dataset and cross-dataset evaluations. Qualitative analysis is provided through t-SNE visualization of the learned feature embeddings.

5.5.1 Quantitative Results

Tables 5.3, 5.4, and 5.5 present face forgery detection performance adapted from standardized DeepfakeBench [8] evaluations. Table 5.3 provides an overview of the detector architectures and their publication venues. All methods are trained on FF++ (HQ) dataset, with intra-dataset evaluation results shown in Table 5.4 and cross-dataset evaluation results in Table 5.5.

Trident achieves state-of-the-art intra-dataset performance with 0.9936 AUC. Cross-dataset results reveal significant performance degradation across all methods. All metrics are obtained through DeepfakeBench's controlled evaluation setting to ensure a fair comparison, and all benchmark results are adapted from [8].

Table 5.3: **Detector Details:** Overview of detector types, backbones, and publication venues.

Type	Detector	Backbone	Venue
Naive	MesoNet [71]	Designed CNN	WIFS-2018
	MesoInception [71]	Designed CNN	WIFS-2018
	CNN-Aug [3]	ResNet [72]	CVPR-2020
	Xception [11]	Xception [9]	ICCV-2019
	EfficientNet-B4 [10]	EfficientNet [10]	ICML-2019
Frequency	F3Net [26]	Xception [9]	ECCV-2020
	SPSL [28]	Xception [9]	CVPR-2021
	SRM [73]	Xception [9]	CVPR-2021
Spatial	Capsule [18]	CapsuleNet [74]	ICASSP-2019
	DSP-FWA [75]	Xception [9]	CVPRW-2019
	Face X-ray [12]	HRNet [76]	CVPR-2020
	FFD [40]	Xception [9]	CVPR-2020
	CORE [41]	Xception [9]	CVPRW-2022
	RECCE [17]	Custom	CVPR-2022
	UCF [45]	Xception [9]	ICCV-2023
	Trident (CNN)	EfficientNet [10]	Ours
	Trident (ViT)	MARLIN [64]	Ours

Table 5.6 presents *Trident*'s detection performance on the FF++ dataset at C40 (LQ) and RAW compression levels. Unlike Table 5.4, we trained and evaluated the models specifically on these compression levels, excluding C23 (HQ).

Table 5.4: **Intra-Dataset Evaluation:** AUC (↑) scores of face forgery detectors on FF++ (HQ) dataset and its subsets.

Type	Detector	FF++ (HQ)	FF-DF	FF-F2F	FF-FS	FF-NT
Naive	MesoNet [71]	0.6077	0.6771	0.6170	0.5946	0.5701
	MesoInception [71]	0.7583	0.8542	0.8087	0.7421	0.6517
	CNN-Aug [3]	0.8493	0.9048	0.8788	0.9026	0.7313
	Xception [11]	0.9637	0.9799	0.9785	0.9833	0.9385
	EfficientNet-B4 [10]	0.9567	0.9757	0.9758	0.9797	0.9308
Frequency	F3Net [26]	0.9635	0.9793	0.9796	0.9844	0.9354
	SPSL [28]	0.9610	0.9781	0.9754	0.9829	0.9299
	SRM [73]	0.9576	0.9733	0.9696	0.9740	0.9295
Spatial	Capsule [18]	0.8421	0.8669	0.8634	0.8734	0.7804
	DSP-FWA [75]	0.8765	0.9210	0.9000	0.8843	0.8120
	Face X-ray [12]	0.9592	0.9794	**0.9872**	0.9871	0.9290
	FFD [40]	0.9624	0.9803	0.9784	0.9853	0.9306
	CORE [41]	0.9638	0.9787	0.9803	0.9823	0.9339
	RECCE [17]	0.9621	0.9797	0.9779	0.9785	0.9357
	UCF [45]	0.9705	**0.9883**	0.9840	0.9896	0.9441
	Trident (CNN)	0.9793	0.9779	0.9843	0.9831	0.9590
	Trident (ViT)	**0.9936**	0.9821	0.9857	**0.9928**	**0.9833**

Table 5.5: **Cross-Dataset Evaluation:** AUC (↑) scores of face forgery detectors across multiple external benchmarks.

Type	Detector	FF++ (LQ)	Celeb-v1	Celeb-v2	DFD	DFDCP	DFDC	UADFV
Naive	MesoNet [71]	0.5920	0.7358	0.6091	0.5481	0.5994	0.5560	0.7150
	MesoInception [71]	0.7278	0.7366	0.6966	0.6069	0.7561	0.6226	0.9049
	CNN-Aug [3]	0.7846	0.7420	0.7027	0.6464	0.6170	0.6361	0.8739
	Xception [11]	0.8261	0.7794	0.7365	0.8163	0.7374	0.7077	0.9379
	EfficientNet-B4 [10]	0.8150	0.7909	0.7487	0.8148	0.7283	0.6955	0.9472
Frequency	F3Net [26]	0.8271	0.7769	0.7352	0.7975	0.7354	0.7021	0.9347
	SPSL [28]	0.8174	**0.8150**	0.7650	0.8122	0.7408	0.7040	0.9424
	SRM [73]	0.8114	0.7926	0.7552	0.8120	0.7408	0.6995	0.9427
Spatial	Capsule [18]	0.7040	0.7909	0.7472	0.6841	0.6568	0.6465	0.9078
	DSP-FWA [75]	0.7357	0.7897	0.6680	0.7403	0.6375	0.6132	0.8539
	Face X-ray [12]	0.7925	0.7093	0.6786	0.7655	0.6942	0.6326	0.8989
	FFD [40]	0.8237	0.7840	0.7435	0.8024	0.7426	0.7029	0.9450
	CORE [41]	0.8194	0.7798	0.7428	0.8018	0.7341	0.7049	0.9412
	RECCE [17]	0.8190	0.7677	0.7319	0.8119	0.7419	0.7133	0.9446
	UCF [45]	0.8399	0.7793	0.7527	0.8074	0.7594	**0.7191**	**0.9528**
	Trident (CNN)	0.7342	0.7609	0.7543	**0.8740**	0.8403	0.6976	0.9450
	Trident (ViT)	**0.8879**	0.7649	**0.8485**	0.8116	**0.8524**	0.6646	0.9323

Table 5.6: *Trident*'s Intra-Dataset **AUC (↑)** Scores on FF++ Dataset at Different Compression Levels

Method	**FF++ (LQ)**	**FF++ (RAW)**
***Trident* (ViT)**	0.9561	0.9951

5.5.1.1 Ablation Studies

Ablation studies were conducted to assess the impact of key components in the *Trident* framework, specifically Triplet Learning (TL), the Gradient Reversal Layer (GRL), the Adversarial Loss (Adv), and the Detached Classification Head (DH). Experiments were performed under two settings:

1. training and testing on all FF++ (HQ) dataset forgery types and
2. training on only *Deepfakes* and *NeuralTextures* manipulations, then testing on all forgery types in the FF++ (HQ) dataset.

The results are presented in Tables 5.7 and 5.8, respectively.

5.5.1.2 Ablation Study on All Forgery Types

Table 5.7 summarizes the model's training and testing results on all forgery types in the FF++ (HQ) dataset. The baseline model (**B**) employs the EfficientNet-B4 backbone with a binary classification head. The variants include the addition of Triplet Learning (**TL**) and the incorporation of the Gradient Reversal Layer (**TL+GRL**).

5.5.1.2.1 Ablation Study on Deepfakes and NeuralTextures In this setting, the models are trained only on *Deepfakes* and *NeuralTextures* manipulations and tested on all forgery types in the FF++ (HQ) dataset. Table 5.8 presents the results. The methods include the baseline (**B**), the addition of Triplet

Table 5.7: Ablation Study on FF++ (HQ) Dataset (All Forgery Types)

Method	AUC (↑)	LogLoss (↓)
Baseline	0.9497	0.2915
TL	0.9613	0.2611
TL+GRL	0.9669	0.2946
TL+GRL+DH	**0.9793**	**0.2456**

Learning (**TL**), the incorporation of the Gradient Reversal Layer (**TL+GRL**), the use of Adversarial Loss (**TL+Adv**), and the inclusion of the Detached Classification Head (**TL+GRL+DH**).

Table 5.8: Ablation Study on FF++ (HQ) Dataset (Deepfakes and NeuralTextures)

Method	AUC (↑)	LogLoss (↓)
Baseline	0.7363	0.9512
TL	0.7506	0.8645
TL+Adv	0.9595	0.3270
TL+GRL	**0.9652**	0.2662
TL+GRL+DH	0.9646	**0.2454**

5.5.1.2.2 Cross-Dataset Ablation on DFD (HQ) To further assess the impact of our proposed components on generalization ability, we conducted an ablation study on the DFD (HQ) dataset with models trained on FF++ (HQ). Table 5.9 presents the results. The *LogLoss* refers to the Binary Cross-Entropy (BCE) loss used for the classification task in both tables.

Table 5.9: Cross-Dataset Ablation on DFD (HQ) Dataset (Training on FF++ (HQ))

Method	AUC (↑)	LogLoss (↓)
Baseline	0.7817	0.6897
TL	0.8438	**0.5522**
TL+GRL	0.8668	0.6076
TL+GRL+DH	**0.8740**	0.5568

5.5.2 Qualitative Results

To qualitatively assess the discriminative capability of the learned embeddings, we employed t-distributed Stochastic Neighbor Embedding (t-SNE) [77] to project high-dimensional features into two-dimensional space.

T-SNE is a nonlinear dimensionality reduction technique widely adopted for visualizing high-dimensional data. Introduced by van der Maaten and Hinton, t-SNE addresses a fundamental challenge in machine learning: interpreting the complex feature spaces that algorithms operate within.

The algorithm converts high-dimensional Euclidean distances between data points into conditional probabilities representing similarities. Points with high similarity in the original space are assigned higher probabilities of being neighbors. t-SNE then constructs a similar probability distribution in the lower-dimensional space and minimizes the Kullback-Leibler divergence between these distributions.

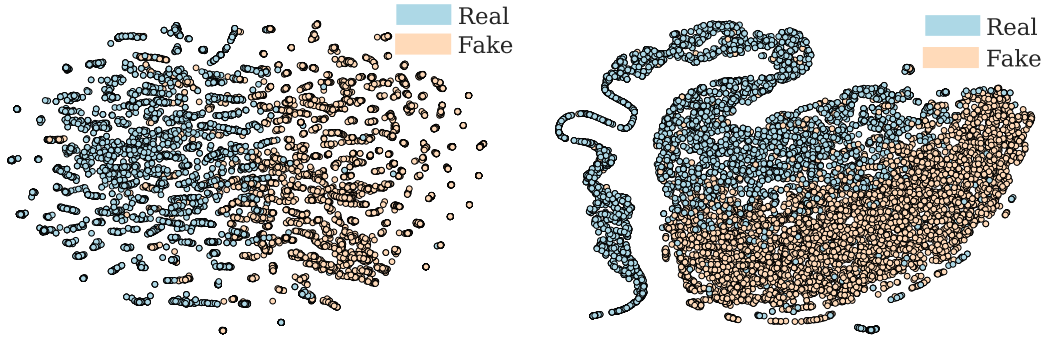
Unlike linear methods such as Principal Component Analysis (PCA), t-SNE preserves local relationships between points, making it particularly useful for revealing cluster structures. Using a Student t-distribution in the low-dimensional space helps address the *crowding problem* often encountered in manifold learning, allowing for a more faithful representation of distant relationships.

To understand how our different model configurations affect feature space organization, we applied t-SNE to the embedding vectors extracted from each model variant. We present two sets of visualizations: baseline models in Figure 5.1 and Triplet Learning-based models in Figure 5.2. Real and fake samples are represented in blue and orange, respectively. All visualizations are produced from the FF++ (HQ) validation split.

Figure 5.1 illustrates embeddings for models *without* Triplet Learning (TL). In Figure 5.1(a) (Baseline, B), real (blue) and fake (orange) samples overlap considerably, indicating weak discriminative power. Adding Gradient Reversal Layer (GRL) and Detached Head (DH) components (B+GRL+DH, Figure 5.1(b))

strengthens real/fake separation but still does not yield five clear clusters for the FF++ forgeries.

In contrast, Figure 5.2 shows *TL-based* models. As seen in Figure 5.2(a) (TL), incorporating TL creates distinct clusters for each forgery type, though some overlap remains. Employing GRL and DH components alongside TL (TL+GRL+DH, Figure 5.2(b)) further separates real from fake while forming five distinct clusters corresponding to the five forgery types present in the FF++ dataset.



(a) Baseline (B) (b) B+GRL+DH

Figure 5.1: t-SNE embeddings for models without TL. GRL+DH strengthens real/fake separation but does not yield five distinct forgery clusters.

(a) TL (b) TL+GRL (c) TL+GRL+DH

Figure 5.2: t-SNE embeddings for TL-based models. TL promotes forgery-type clustering; GRL and DH enhance real/fake separation.

Figure 5.3 shows the embeddings produced by a ViT-based model (MARLIN). Compared to previous configurations, MARLIN offers even more distinctive separation across forgery clusters and isolates pristine samples from fake ones.

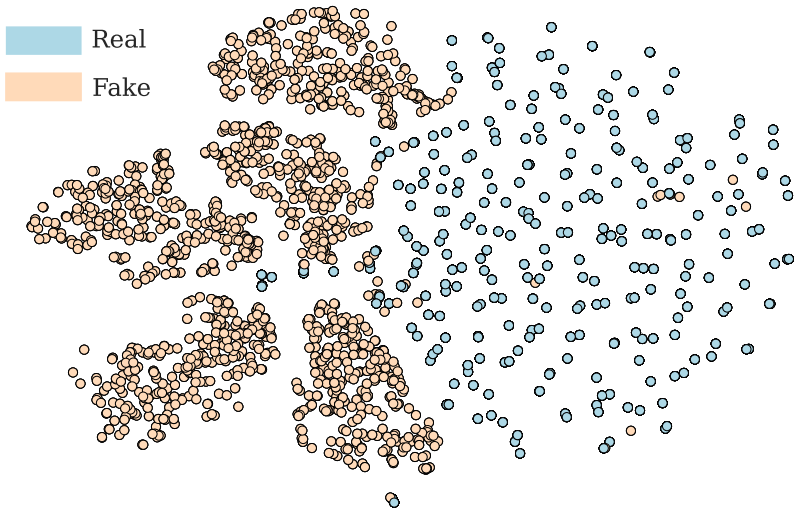


Figure 5.3: t-SNE embeddings from the MARLIN (ViT) backbone. This model exhibits improved clustering among the five FF++ forgeries and more pronounced real/fake separation.

5.6 Discussion

The experimental results presented in Section 5.5 provide several important insights into face forgery detection:

5.6.1 Performance Comparison

The evaluation results in Tables 5.4 and 5.5 demonstrate *Trident* (ViT) reaches 0.9936 AUC on FF++ (HQ), surpassing the previous SOTA approach UCF (0.9705). In cross-dataset evaluation, *Trident* variants significantly outperform existing methods on 4 out of 7 benchmarks, with significant improvements on FF++ (LQ) (0.8879 vs. UCF's 0.8399), Celeb-v2 (0.8485 vs. SPSL's 0.7650), DFD (0.8740 vs. Xception's 0.8163), and DFDCP (0.8524 vs. UCF's 0.7594). On three benchmarks—Celeb-v1, DFDC, and UADFV—other methods achieve higher AUC scores: SPSL (0.8150), UCF (0.7191 and 0.9528), respectively. This

mixed performance across different datasets reveals the complexity of cross-domain forgery detection and highlights our approach's strengths and limitations.

5.6.2 Component Contributions

The ablation studies quantify each component's impact. From Tables 5.7 and 5.8, Triplet Learning provides a foundational improvement, raising the baseline AUC from 0.9497 to 0.9613 with all forgery types and from 0.7363 to 0.7506 with limited forgery types. The most significant gain occurs when combining TL with adversarial techniques (GRL/Adv): when trained only on Deepfakes and NeuralTextures but tested on all forgery types, TL+GRL achieves 0.9652 AUC—a 22.89% increase over the baseline's 0.7363. This demonstrates the framework's ability to learn forgery-agnostic features. The cross-dataset results in Table 5.9 reinforce this finding, illustrating TL+GRL+DH achieves 0.8740 AUC on DFD, surpassing the baseline by 9.23%.

5.6.3 Feature Space Organization

The t-SNE visualizations in Figures 5.1 and 5.2 reveal the models' discriminative capabilities. TL-based models produce more distinct clustering by forgery type, with the full *Trident* configuration (TL+GRL+DH) achieving clear real/fake separation and distinct clustering for different forgery techniques. Figure 5.3 demonstrates that the ViT backbone further enhances these separation characteristics with more pronounced boundaries between clusters.

5.6.4 CNN vs. ViT Backbone Comparison

The results reveal distinct performance patterns between backbone architectures. ViT-based *Trident* achieves higher performance in intra-dataset evaluation (0.9936 AUC) and on specific cross-dataset benchmarks (FF++ LQ: 0.8879,

Celeb-v2: 0.8485, DFDCP: 0.8524). While CNN-based *Trident* yields better results on DFD (0.8740 vs. 0.8116), DFDC (0.6976 vs. 0.6646), and UADFV (0.9450 vs. 0.9323), the ViT architecture demonstrates overall superior performance across the majority of benchmarks. This performance advantage can be attributed to the ViT architecture's self-attention mechanism, which appears particularly effective for modeling complex forgery artifacts in both intra-dataset and cross-domain evaluations.

5.6.5 Generalization Challenges

Despite improvements achieved, cross-dataset performance remains significantly lower than intra-dataset results. The marked performance degradation on benchmarks like DFDC, where all methods achieve below 0.72 AUC, highlights persistent generalization challenges. The variable performance across different test sets suggests current deepfake detection methods remain sensitive to the specific characteristics of forgery types, compression artifacts, and dataset capture conditions. These observations point to fundamentally challenging aspects of domain generalization in forgery detection that merit further investigation.

5.6.6 Implications for Face Forgery Detection

Integrating triplet learning and adversarial components in the *Trident* framework enhances deepfake detection models' discriminative power and generalization ability. The results suggest that adversarial training is particularly effective in preventing overfitting and promoting robustness to various forgery types, especially when the training data is limited. The combination of triplet learning with adversarial approaches leads to significant performance improvements, indicating that these components are complementary in addressing the challenges of deepfake detection.

Chapter 6

Conclusion

Deep learning models have demonstrated remarkable success across diverse domains, yet they frequently encounter challenges adapting to *unseen* or *shifting* conditions. This phenomenon is often referred to as *domain shift*: when a model trained on one data distribution must generalize to data points that differ in fundamental, unrepresented ways. In the context of face forgery detection, domain shift can be pronounced: new manipulation techniques continually evolve, with subtle, hard-to-detect artifacts. Consequently, models trained on a fixed set of known forgeries often fail to maintain high accuracy against novel manipulations, highlighting the pressing need for flexible and robust detection strategies.

Metric learning has emerged as a promising approach to address these limitations. Instead of relying solely on traditional classification, metric learning attempts to build a representation space where meaningful similarities and differences between samples can be easily discerned. *Triplet learning* is particularly relevant here, as it enforces proximity between anchor-positive pairs while maximizing the distance to negative samples. When applied to face forgery detection, triplet learning directs the model to focus on those subtle artifacts introduced by manipulation while controlling for potentially confounding factors such as identity, background, or pose, yielding embeddings that can better detect previously unseen forgeries.

Nonetheless, developing embeddings that remain effective across various manipulation strategies is nontrivial. This challenge is where *adversarial training* proves considerably useful. By training an embedding generator in opposition to a forgery discriminator that aims to classify specific forgery techniques, the model is compelled to eliminate forgery-type-specific cues, honing in on generalizable signals of fakery. Pairing adversarial training with triplet learning enables the network to learn a more holistic, manipulation-agnostic representation space, thereby boosting its ability to detect new or evolving forgeries without direct retraining on those methods.

Guided by these insights, we introduced *Trident*, a face forgery detection framework that unifies triplet learning and adversarial training to counter the increasing sophistication and variety of forgery methods. Central to *Trident* is the curation of triplets that share identity and setting, but differ in authenticity, effectively highlighting manipulation-induced discrepancies over incidental variations. This design compels the model to concentrate on genuinely discriminative features and fosters the learning of robust embeddings, even when the forgery technique deviates significantly from those observed in training.

Simultaneously, the adversarial interaction with a dedicated forgery discriminator encourages *forgery-agnostic* embeddings. Rather than capturing artifacts tied to a specific forgery style, the learned features encode broader signals of fakery. We observed tangible performance gains in our experiments—including higher accuracy, increased AUC, and more reliable detection across various datasets—once we integrated the gradient reversal layer, triplet loss, and selective backpropagation blocking. Notably, training on a diverse assortment of forgery types consistently yielded strong results, while cross-evaluation on unseen datasets underscored *Trident's* improved capacity to generalize, thereby addressing a significant limitation commonly seen in existing supervised approaches.

Experiments demonstrate that *Trident* achieves state-of-the-art intra-dataset

performance and outperforms existing methods on 4 out of 7 cross-dataset benchmarks. Our ablation studies confirm combining triplet learning, gradient reversal layers, and strategic backpropagation blocking significantly enhances detection performance. The t-SNE visualizations further validate our approach by clearly separating real and fake samples. While *Trident* demonstrates improved cross-dataset performance, the gap between intra-dataset and cross-dataset results highlights persistent generalization challenges in face forgery detection.

There remain several paths for further exploration. Incorporating temporal information could assist in identifying subtle inconsistencies across consecutive video frames while leveraging multi-modal data (e.g., audio tracks, text metadata) might enhance detection performance in multi-faceted real-world scenarios.

In addition, adopting dynamic sampling or margin-tuning strategies within the triplet loss framework could yield a more discriminative embedding space, and self-supervised or semi-supervised pretraining might reduce reliance on large labeled datasets. Pursuing these directions could further refine and extend *Trident* to create an even more robust, flexible, and future-proof solution for face forgery detection.

Bibliography

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, NIPS '14, pp. 2672–2680, 2014. 1, 8

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33 of *NIPS '20*, pp. 6840–6851, 2020. 1

[3] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. 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 '20, pp. 8695–8704, 2020. 1, 41, 42

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pp. 815–823, 2015. 1, 8, 19

[5] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1180–1189, PMLR, 07–09 Jul 2015. 2, 9, 17

[6] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 of *CVPR '05*, pp. 539–546, IEEE, 2005. 6, 19

[7] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "Siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, vol. 6 of *NIPS '93*, 1993. 7, 19

[8] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "DeepfakeBench: A comprehensive benchmark of deepfake detection," in *Advances in Neural Information Processing Systems*, vol. 36 of *NIPS '23*, pp. 4534–4565, 2023. 11, 41

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '17, pp. 1251–1258, 2017. 11, 20, 41

[10] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning*, ICML '19, pp. 6105–6114, 2019. 11, 20, 39, 41, 42

[11] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '19, pp. 1–11, 2019. 11, 27, 35, 37, 38, 41, 42

[12] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face x-ray for more general face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '20, pp. 5001–5010, 2020. 11, 41, 42

[13] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, (Berlin, Heidelberg), p. 103–120, Springer-Verlag, 2020. 12

[14] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 2185–2194, 2021. 12

[15] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 6111–6121, 2022. 12

[16] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '22, pp. 18720–18729, 2022. 12

[17] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang, "End-to-end reconstruction-classification learning for face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '22, pp. 4113–4122, 2022. 12, 41, 42

[18] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '19, pp. 2307–2311, 2019. 12, 41, 42

[19] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proceedings of the IEEE 10th International Conference on Biometrics Theory, Applications and Systems*, BTAS '19, pp. 1–8, 2019. 12

[20] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '19, pp. 8261–8265, 2019. 12, 38

[21] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, "Exploring temporal coherence for more general video face forgery detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '21, pp. 15044–15054, 2021. 12

[22] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 5039–5049, 2021. 12

[23] I. Choi, Y. Kim, and S. Hwang, "Exploiting inconsistencies in stylegan latent space for deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '24, 2024. 12

[24] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, F. Huang, and L. Ma, "Spatiotemporal inconsistency learning for deepfake video detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 10343–10352, 2021. 12

[25] I. Amerini, R. Caldelli, and F. Picchioni, "Deepfake video detection through optical flow based CNN," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, ICCVW '19, pp. 1205–1207, 2019. 13

[26] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, (Berlin, Heidelberg), p. 86–103, Springer-Verlag, 2020. 13, 41, 42

[27] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 6458–6467, 2021. 13

[28] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 772–781, 2021. 13, 41, 42

[29] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, pp. 3247–3258, JMLR.org, 2020. Article no. 304. 13

[30] U. A. Ciftci, I. Demir, and L. Yin, "How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 3066–3078, 2020. 13

[31] K. Stefanov, B. Paliwal, and A. Dhall, "Visual representation of physiological signals for fake video detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '22, pp. 9970–9979, 2022. 14

[32] Z. Wang, Z. Cheng, J. Xiong, X. Xu, T. Li, B. Veeravalli, and X. Yang, "A timely survey on vision transformer for deepfake detection," 2024. Available at `https://arxiv.org/abs/2405.08463`, Accessed: 2025-03-27. 14

[33] C. Zhao *et al.*, "ISTVT: Interpretable spatial-temporal video transformer for deepfake detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1335–1348, 2023. 14

[34] J. Wang, Z. Wu, J. Chen, and Y. Jiang, "M2TR: Multi-modal multi-scale transformers for deepfake detection," *CoRR*, vol. abs/2104.09770, 2021. Available at `https://arxiv.org/abs/2104.09770`, Accessed: 2025-03-27. 14

[35] X. Dong, J. Bao, D. Chen, T. Zhang, W. Zhang, N. Yu, D. Chen, F. Wen, and B. Guo, "Protecting celebrities from deepfake with identity consistency transformer," 2022. Available at `https://arxiv.org/abs/2203.01318`, Accessed: 2025-03-27. 14

[36] W. Zhuang, Q. Chu, Z. Tan, Q. Liu, H. Yuan, C. Miao, Z. Luo, and N. Yu, "UIA-ViT: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection," 2022. Available at `https://arxiv.org/abs/2210.12752`, Accessed: 2025-03-27. 14

[37] S. A. Khan and H. Dai, "Video transformer for deepfake detection with incremental learning," *CoRR*, vol. abs/2108.05307, 2021. Available at `https://arxiv.org/abs/2108.05307`, Accessed: 2025-03-27. 14

[38] D. Wodajo, S. Atnafu, and Z. Akhtar, "Deepfake video detection using generative convolutional vision transformer," 2023. Available at `https://arxiv.org/abs/2307.07036`, Accessed: 2025-03-27. 14

[39] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to GANs: Learning and analyzing GAN fingerprints," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '19, pp. 7556–7566, 2019. 15

[40] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '20, pp. 5781–5790, 2020. 15, 41, 42

[41] Y. Ni, D. Meng, C. Yu, C. Quan, D. Ren, and Y. Zhao, "CORE: Consistent representation learning for face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '22, pp. 12–21, 2022. 15, 41, 42

[42] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '21, pp. 15023–15033, 2021. 15

[43] X. Huang, F. Xue, B. Fan, L. Zhong, Y. Fu, and Q. Tian, "Implicit identity leakage: The stumbling block to improving deepfake detection generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '23, pp. 3994–4004, 2023. 15

[44] Y. Zhai, D. Fan, G. Chen, P. Wei, M.-M. Cheng, and L. Shao, "Towards weakly supervised general image forgery detection and localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '23, 2023. 15

[45] Z. Yan, Y. Zhang, Y. Fan, and B. Wu, "UCF: Uncovering common features for generalizable deepfake detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '23, pp. 22355–22366, 2023. 15, 41, 42

[46] U. Ojha, Y. Li, and Y. J. 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 '23, pp. 24480–24489, 2023. 15

[47] T. Reiss, B. Cavia, and Y. Hoshen, "Detecting deepfakes without seeing any," 2023. Available at `https://arxiv.org/abs/2311.01458`, Accessed: 2025/03/27. 15

[48] C. Tan, Y. Zhao, S. Wei, G. Gu, and Y. Wei, "Learning on gradients: Generalized artifacts representation for gan-generated images detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '23, pp. 12105–12114, 2023. 15

[49] S. Fung, X. Lu, C. Zhang, and C. Li, "DeepfakeUCL: Deepfake detection via unsupervised contrastive learning," *CoRR*, vol. abs/2104.11507, 2021. Available at `https://arxiv.org/abs/2104.11507`, Accessed: 2025/03/27. 16

[50] Z. Sun, S. Chen, T. Yao, B. Yin, R. Yi, S. Ding, and L. Ma, "Contrastive pseudo learning for open-world deepfake attribution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '23, pp. 20825–20835, 2023. 16

[51] Y. Xu, K. Raja, and M. Pedersen, "Supervised contrastive learning for generalizable and explainable deepfakes detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, WACVW '22, pp. 379–389, 2022. 16

[52] C.-Y. Hong, Y.-C. Hsu, and T.-L. Liu, "Contrastive learning for DeepFake classification and localization via multi-label ranking," in *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR '24, pp. 17627–17637, 2024. 16

[53] Z. Gu, T. Yao, Y. Chen, S. Ding, and L. Ma, "Hierarchical contrastive inconsistency learning for deepfake video detection," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, (Berlin, Heidelberg), pp. 596–613, Springer-Verlag, 2022. 16

[54] N. Larue, N.-S. Vu, V. Struc, P. Peer, and V. Christophides, "SeeABLE: Soft discrepancies and bounded contrastive learning for exposing deepfakes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV '23, pp. 20954–20964, IEEE, 2023. 16

[55] B. Liang, Z. Wang, B. Huang, Q. Zou, Q. Wang, and J. Liang, "Depth map guided triplet network for deepfake face detection," *Neural Networks*, vol. 159, pp. 34–42, 2023. 16

[56] A. Kumar, A. Bhavsar, and R. Verma, "Detecting deepfakes with metric learning," in *Proceedings of the 8th International Workshop on Biometrics and Forensics*, IWBF '20, pp. 1–6, 2020. 16

[57] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pp. 2823–2832, 2020. 16

[58] N. Beuve, W. Hamidouche, and O. Déforges, "Hierarchical learning and dummy triplet loss for efficient deepfake detection," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 20, no. 3, 2024. Article No. 89, 18 pages. 16

[59] A. Jain, P. Korshunov, and S. Marcel, "Improving generalization of deepfake detection by training for attribution," in *Proceedings of the IEEE 23rd International Workshop on Multimedia Signal Processing*, MMSP '21, pp. 1–6, 2021. 16

[60] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "Eann: Event adversarial neural networks for multi-modal fake news detection," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 849–857, 2018. 17

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017. 21

[62] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021. Available at `https://openreview.net/forum?id=YicbFdNTTy`, Accessed: 2025/03/27. 22

[63] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '22, pp. 16000–16009, June 2022. 24

[64] Z. Cai, S. Ghosh, K. Stefanov, A. Dhall, J. Cai, H. Rezatofighi, R. Haffari, and M. Hayat, "Marlin: Masked autoencoder for facial video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '23, pp. 1493–1504, 2023. 25, 39, 41

[65] J. Wang, Y. Liu, Y. Hu, H. Shi, and T. Mei, "FaceX-Zoo: A PyTorch toolbox for face recognition," in *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pp. 3779–3782, 2021. 31

[66] Google and Jigsaw, "DeepFake detection dataset," 2019. Available at `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`, Accessed: 2025-03-27. 35, 38

[67] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '20, pp. 3207–3216, 2020. 37, 38

[68] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (DFDC) dataset," *arXiv preprint arXiv:2006.07397*, 2020. Available at `https://arxiv.org/abs/2006.07397,Accessed:2025-03-27`. 38

[69] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," *arXiv preprint arXiv:1910.08854*, 2019. Available at `https://arxiv.org/abs/1910.08854,Accessed:2025-03-27`. 38

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations*, ICLR '15, 2015. Available at `https://arxiv.org/abs/1412.6980`, Accessed: 2025-03-27. 40

[71] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proceedings of the IEEE International Workshop on Information Forensics and Security*, WIFS '18, pp. 1–7, IEEE, 2018. 41, 42

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pp. 770–778, 2016. 41

[73] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '21, pp. 16317–16326, 2021. 41, 42

[74] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, vol. 30 of *NIPS '17*, pp. 3859–3869, 2017. 41

[75] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, CVPRW '19, pp. 46–52, June 2019. 41, 42

[76] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020. 41

[77] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, Nov 2008. 45