Nuisance-Aware Representation Refinement for Robust Deepfake Detection

Vinayak Mishra

***Abstract*—Deepfake detectors often rely on dataset-specific artifacts, limiting robustness under compression and cross-dataset evaluation. We propose NARR, a Nuisance-Aware Representation Refinement module designed as a lightweight plug-in for CNN backbones. NARR operates directly on convolutional feature maps using multi-scale nuisance estimation and dual-channel spatial gating to suppress unstable pattedual-channel erving discriminative cues.**

To evaluate robustness, we compare NARR against nine recent deepfake detection approaches under a unified training protocol. Lightweight approximations of each method are reimplemented to enable comparison under equivalent model complexity; results reflect relative behavior under identical conditions rather than original paper performance. All models are trained on FaceForensics++ and evaluated across JPEG compression levels and cross-dataset settings on DFDC and Celeb-DF. Experiments show that nuisance-aware refinement improves stability under distribution shifts while introducing minimal computational overhead. Code and complete benchmark logs are available at the GitHub link.

***Index Terms*—Deepfake Detection, Generalization, CNN, Transformer, Representation Learning**

I. Introduction

Deepfake detection has advanced with CNN and transformer architectures, yet many models struggle to generalize under distribution shifts such as heavy compression and cross-dataset evaluation. A common failure mode is reliance on *nuisance patterns* which are dataset-specific artifacts (e.g., codec traces or compression fingerprints) that do not correspond to true manipulation cues.

We address this problem using a modular refinement strategy that can be attached to existing backbones. NARR (Nuisance-Aware Representation Refinement) is a lightweight plug-in operating on convolutional feature maps. By preserving spatial structure and avoiding global pooling, NARR estimates nuisance signals through multi-scale convolutions and applies channel-wise and spatial gating to refine representations.

We evaluate NARR alongside nine recent deepfake detection approaches under a unified protocol. Lightweight approximations of each method are reimplemented and trained for five epochs on FaceForensics++ to enable fair comparison under similar model complexity. Reported results, therefore, reflect relative behavior under identical conditions rather than original paper performance.

All models are e,valuated ,on FF++ test data, JPEG compression levels, and cross-dataset benchmarks (DFDC and Celeb-DF) without fine-tuning. Results indicate that nuisance-aware refinement improves robustness to compression and dataset shifts.

The contributions of this work are as follows:

- **NARR module:** A lightweight nuisance-aware refinement layer for CNN feature maps.
- **Controlled evaluation:** Comparison against recent detectors under a unified training protocol.
- **Robustness analysis:** Evaluation across compression stress and cross-dataset testing.
- **Plug-in design:** Minimal architectural changes with low computational overhead.

II. Related Work

This section briefly summarizes the main categories of deepfake detection methods included in our benchmark. The goal is not to provide an exhaustive review, but to group the evaluated papers by their general design direction.

A. Vision–Language Model Based Methods

Some recent detectors use large vision-language models, such as CLIP, for deepfake detection. These approaches often focus on transferability and cross-domain robustness. The benchmark includes:

- CLIPping the Deception: Adapting Vision-Language Models for Universal Deepfake Detection
- Unlocking the Hidden Potential of CLIP in Generalizable Deepfake Detection
- Rethinking Vision-Language Model in Face Forensics: Multi-Modal Interpretable Forged Face Detectoruse2-Det) vision-languageics Ada,pter: Adaptin,g CLIP for Generalizable Face Forgery Detection

B. Representation and Reconstruction Based Methods

Another group of methods focuses on learning manipulation-sensitive features through representation learning or reconstruction-based objectives. The benchmark includes:

- Real Face Foundation Representation Learning (RFFR)
- Contrasting Deepfakes Diffusion via Contrastive Learning and Global-Local Similarities (CoDE)

C. Generalization and Optimization Driven Methods

Some works directly target cross-dataset robustness using specialized training objectives or optimization strategies. The benchmark includes:

- RoGA: Towards Generalizable Deepfake Detection through Robust Gradient Alignment

- Preserving Fairness Generalization in Deepfake Detection
- D^3: Scaling Up Deepfake Detection by Learning from Discrepancy

D. Positioning of NARR

Unlike these approaches, which mainly introduce new backbones or training objectives, NARR is designed as a lightweight refinement module that can be added to existing CNN detectors. The unified benchmark allows us to compare this plug-in design against a range of existing model families under the same training setup.

III. Unified Benchmark Protocol

To ensure fair and reproducible comparisons, we adopt a unified benchmark protocol that standardizes (i) training data, (ii) preprocessing, (iii) evaluation splits, and (iv) robustness stress tests.

A. Datasets and Splits

Training is performed exclusively on FaceForensics++ (FF++) using a binary real/fake protocol. Evaluation includes FF++ in-domain testing, compression robustness on recompressed FF++ test frames, and cross-dataset testing on DFDC and Celeb-DF without fine-tuning.

B. Preprocessing and Frame Sampling

All datasets are represented as frame-level face crops to standardize the input format across training and evaluation. For video datasets, we sample $K = 16$ frames per video using uniform indexing across the full video duration. Each sampled frame is converted to RGB and passed through an MTCNN face detector; when a face is detected, we export the athe ligned face crop at 224×224 resolution. Frames with no detected face are skipped. Crops are stored as JPEG images (quality 95) in class-labeled folders (`real`/`fake`).

For FF++, we construct deterministic train/val/test splits from the provided metadata (stratified by label) with a 70/15/15 ratio. For Celeb-DF, we follow the official test list and split the remaining videos into train/val (85/15, stratified). DFDC evaluation uses the prepared validation face-crop image folders in the same 224×224 format.

During training, we apply lightweight data augmentation (random affine perturbations, color jitter, blur, grayscale, gamma adjustment, sharpening, and horizontal flip), while evaluation uses only deterministic resizing and tensor conversion. For JPEG robustness, we additionally recompre,ss FF++ test frames at the target quality level before resizing and evaluation.

C. Metrics

We emphasize AUC and Accuracy (ACC) as the primary metrics reported in this paper, since AUC is less sensitive to threshold choice and ACC provides an intuitive operating-point summary. Full metric breakdowns (including Precision, Recall, and F1) and complete evaluation logs are available in the accompanying GitHub repository.

D. Robustness and Generalization Tests

Compression robustness evaluates models on FF++ test frames recompressed with JPEG quality levels $\{100, 90, 75, 50, 30\}$. **Cross-dataset generalization** evaluates the same FF++-trained models on DFDC and Celeb-DF with no adaptation.

IV. Proposed Method (NARR)

A. Overview

We propose NARR (Nuisance-Aware Representation Refinement), a modular feature refinement layer designed to be attached to the output of any convolutional backbone without modifying its internal architecture. Given convolutional feature maps extracted from a CNN, NARR estimates nuisance patterns and adaptively refines representations before downstream embedding and classification.

An input image is processed by a CNN backbone, followed by the proposed NARR module, token embedding, a transformer encoder, and a lightweight classifier. The overall pipeline is illustrated in Fig. 1.

Unlike many deepfake detectors that rely on global pooling, spatial feature maps are preserved to enable localized nuisance suppression. NARR (Ours) 0.9378 0.8533 0.7052 0.7113 Baseline (no NARR) 0.9331 0.8467 0.6548 0.6837

B. Backbone Feature Extraction

Let $F \in \mathbb{R}^{C \times H \times W}$ denote the convolutional feature maps extracted from a backbone network. The backbone can be any CNN architecture; in our im,plementation a ResNet-based encoder is used. Global pooling is intentionally removedthe to preserve spatial structure required for refinement.

C. Multi-scale Nuisance Estimation

Dataset-specific artifacts appear at different spatial scales. To capture these patterns, we estimate a nuisance representation $\hat{N}$ using parallel convolutions:

$$\hat{N} = \phi([f_{1\times1}(F), f_{3\times3}(F), f_t hedil(F)]) \tag{1}$$

where $f_{1\times1}$ captures channel-wise artifacts, $f_{3\times3}$ ca,ptures mid-scale patterns, and f_{dil} denotes dilated convolutions modeling large-scale structures. The outputs are concatenated and projected back to the original channel dimension.

D. Channel and Spatial Gating

NARR generates two complementary gating masks:

$$G_c = \sigma(\text{Conv}_{1\times1}(\text{GAP}(\hat{N}))), \quad G_s = \sigma(\text{Conv}_{1\times1}(\hat{N})) \tag{2}$$

where G_c captures channel importance and G_s captures spatial relevance. The final nuisance confidence map is defined as

$$G = G_c \cdot G_s. \tag{3}$$

dual-gatingating mechanism enathe bles selective suppression of nuisance-heavy regions while preserving informative features.

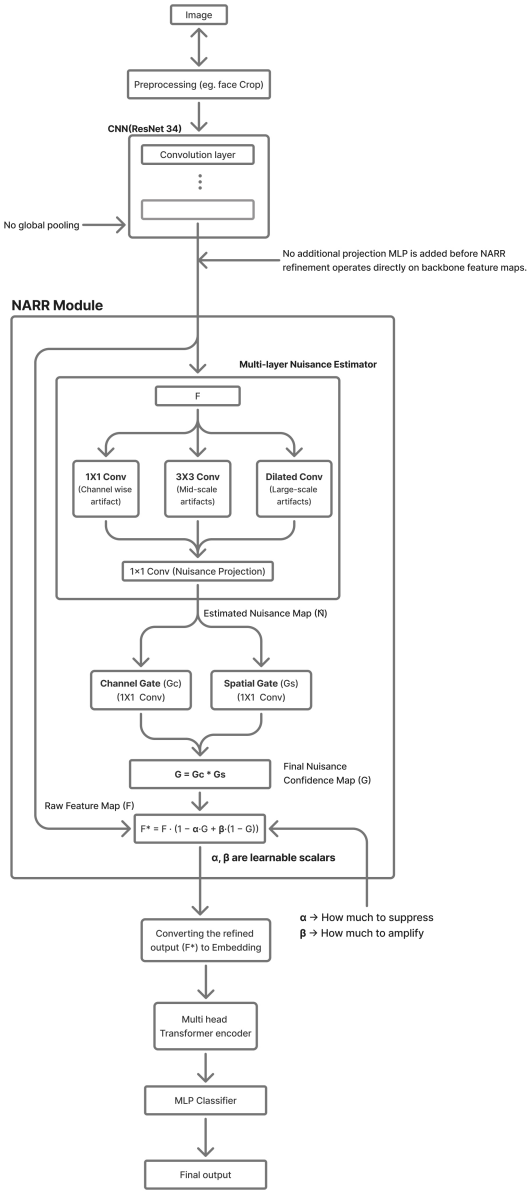

Fig. 1. Overview of the proposed NARR pipeline (CNN backbone + NARR + tokenization + transformer + classifier).

E. Didual-gatingom Classical Attentthe ion Mechanisms

Unlike conventional attention or gating mechanisms that directly learn importance weights from task-discriminative features, NARR explicitly models nuisance structure as a separate intermediate represen,tation before feature modulation. Standard attention modules typically amplify salient regions correlated with classification objectives, which can unintentionally reinforce dataset-s,pecific artifacts or compression fingerprints. In contrast, NARR first estimates a nuisance map through multi-scale convolutions designed to capture instability patterns rather than semantic importance. The subsequent dual-channel gating operates on this nuisance estimate, enabling suppression of distribution-dependent signals while preserving discriminative content. This design shifts the objective from feature selection to representation refinement, where modulation is guided by predicted nuisance confidence rather than learned attention weights. As a result, NARR behaves fundamentally differently from classical spatial or channel attention blocks: instead of highlighting informative regions, it aims to attenuate features that reduce cross-domain stability, making the module particularly suited for robustness under compression and dataset shift.

F. Feature Refinement

Given nuisance confidence G, refined features are computed as:

$$F^* = F \cdot (1 - \alpha G + \beta(1 - G)) \tag{4}$$

where α and β are learnable scalars controlling suppression and amplification respectively. This formulation allows the network to attenuate noisy patterns while reinforcing useful signals without altering backbone parameters.

G. Token Embedding and Classification

Refined feature maps are projected into token embeddings using a 1×1 convolution followed by adaptive pooling. The resulting tokens are processed by a multi-head transformer encoder to capture global relationships before final classification.

H. Training Objectives

The model is optimized using binary classification loss together with a nuisance-invariance objective that encourages consistent nuisance representations across corrupted views:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_{inv}\mathcal{L}_{inv} + \lambda_{dom}\mathcal{L}_{dom}. \tag{5}$$

The invariance term is computed via contrastive similarity between representations extracted from clean and corrupted samples, improving robustness to compression and noise. We additionally include a lightweight domain-adversarial objective $\mathcal{L}_{dom}$ to encourage invariance across corruption domains under the unified protocol.

I. Implementation Details

NARR is implemented as a lightweight module appended to the end of a CNN backbone. Because it operates only on feature tensors, no modifications to backbone training or architecture are required, making it suitable as a drop-in enhancement for existing detectors.

J. Expected Impact on Robustness

By explicitly refining nuisance features instead of modifying backbone structure, NARR targets robustness to distribution shifts such as strong compression and cross-dataset evaluation. NARR maintains high in-domain performance on FF++ and remains competitive under aggressive JPEG compression. On DFDC, NARR achieves the strongest AUC in our benchmark

TABLE I
ARCHITECTURE COMPONENTS

Component	Description	Params
CNN Backbone	Frozen ResNet-34 feature extractor without global pooling	Pretrained
NARR Module	Nuisance-aware feature refinement layer	Lightweight
Tokenization	Spatial features reshaped into tokens	$512 \to D$
Transformer Encoder	2-layer transformer, 4 heads, FFN=512	Learnable
Classifier (MLP)	Linear head with token mean pooling	$D \to 1$

under the same training budget, suggesting that nuisance-aware refinement can improve stability without increasing architectural complexity.

TABLE II
HIGH-LEVEL AUC SUMMARY

Model	FF++ AUC	JPEG30 AUC	DFDC AUC	CelebDF AUC
Baseline (no NARR)	0.9331	0.8467	0.6548	0.6837
NARR (Ours)	0.9378	0.8533	0.7052	0.7113

V. EXPERIMENTS

This section presents a unified benchmarking study and a focused evaluation of NARR under distribution shifts. We compare nine representative deepfake detection approaches (ten models including NARR), where each baseline is a lightweight implementation of the original method trained under the same protocol. The goal is controlled relative comparison under identical conditions rather than reproducing the original papers' absolute numbers. In addition to summary tables, complete benchmark statistics (ACC, AUC, Precision, Recall, F1) and evaluation logs for all models are released in the GitHub repository.

A. Training Setup

All models are trained under identical conditions to ensure fairness. Images are resized to 224×224 and optimized using Adam with a learning rate of 1×10^{-4}. The batch size is set to 16 and models are trained for 5 epochs.

Data augmentation includes geometric perturbations, color jitter, grayscale conversion, blur, gamma adjustment, and random sharpening. Additional corruption-based augmentations such as spatial degradation and frequency perturbation are applied to encourage nuisance invariance. Batch normalization layers inside the backbone remain frozen to stabilize training. This configuration is shared across NARR and all lightweight benchmark implementations.

B. Computational Cost Analysis

A key design objective of NARR is to improve robustness without substantially increasing computational complexity. We compare the baseline detector (CNN backbone + token embedding + transformer classifier) with the NARR-enhanced architecture using an input resolution of 224×224. Parameter count and floating-point operations (FLOPs) are measured using a single forward pass with batch size 1.

TABLE III
TRAINING CONFIGURATION

Parameter	Value
Input Resolution	224×224
Batch Size	16
Optimizer	Adam
Learning Rate	1×10^{-4}
Epochs	5
Loss	BCE + Invariance Loss

TABLE IV
COMPUTATIONAL COST COMPARISON BETWEEN THE BASELINE DETECTOR AND THE PROPOSED NARR MODULE.

Model	**Params (M)**	**GFLOPs**	Δ GFLOPs
Baseline Detector	21.94	3.69	–
NARR (Ours)	23.65	3.76	+1.9%

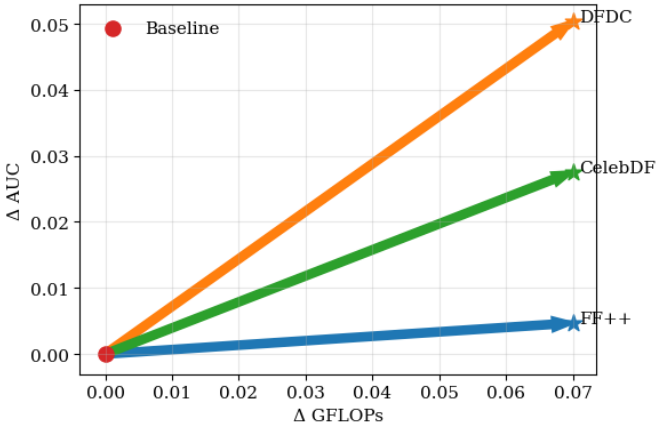

Fig. 2. Compute–robustness trade-off under the unified protocol. Plot AUC versus GFLOPs (and/or Params) to visualize NARR's overhead relative to robustness gains.

Despite introducing additional nuisance estimation and dual-gating operations, NARR increases computational cost only marginally. While parameter count grows by approximately 7.8%, the forward-pass computation increases by only 1.9% compared to the baseline architecture. This modest overhead arises because NARR operates directly on intermediate feature maps using lightweight convolutional projections rather than expanding backbone depth or transformer capacity. These results support the claim that NARR functions as a lightweight plug-in module, enabling improved robustness under compression and cross-dataset evaluation with minimal additional computation.

C. Unified Benchmark Models

To enable controlled comparison, we implement lightweight approximations of nine recent deepfake detection methods mentioned in the Related Work section (resulting in ten models including NARR). Each method is trained under the same preprocessing pipeline, optimizer configuration, and number of epochs. Results therefore reflect relative behavior under identical conditions rather than original paper performance.

Full model specifications, full metric tables, and complete evaluation outputs for all benchmarked models are provided in the GitHub repository. Here we focus on the main approaches and the most notable results.

TABLE V
BENCHMARK METHODS UNDER UNIFIED PROTOCOL

Method	Family	Main Idea
NARR (Ours)	Refinement plug-in	Nuisance-aware feature refinement
Baseline Detector	CNN + Transformer	Baseline backbone + tokenization + classifier
RFFR	Representation	Real-face foundation representation learning
CLIPping the Deception	Vision–language	CLIP adaptation for deepfake detection
Unlocking CLIP	Vision–language	CLIP-based generalizable detection
RoGA	Optimization	Robust gradient alignment
Forensics Adapter	Vision–language	Adapter-based CLIP for forensics
D^3	Generalization	Learning from discrepancy
Frequency-Aware	Frequency	Frequency-space learning
ForgeLens	Data-efficient	Forgery-focused learning
Xception Detector	CNN	Xception-based detector

D. Benchmark Results

We first report in-domain performance on the FF++ test split. Table VI summarizes Accuracy and AUC. For completeness, full metric breakdowns (Precision, Recall, F1) and evaluation logs are reported in the GitHub repository.

TABLE VI
FF++ TEST RESULTS

Model	ACC	AUC
NARR (Ours)	0.8671	0.9378
Baseline (no NARR)	0.8509	0.9331
RFFR	0.8673	0.9309
Unlocking CLIP	0.8839	0.9386
CLIPping the Deception	0.8085	0.8221
RoGA	0.7458	0.8445
Forensics Adapter	0.7621	0.7876
D^3	0.8276	0.8480
Frequency-Aware	0.7210	0.6729
ForgeLens	0.7929	0.5861
Xception Detector	0.7965	0.8431

E. Ablation Study: Effect of NARR

To isolate the contribution of the proposed refinement module, we construct an ablation baseline that matches NARR's detector family and training protocol but removes the NARR block. Concretely, the ablation baseline uses the same CNN backbone, tokenization, transformer encoder, and classifier as NARR. It keeps the same corruption-based invariance training and domain-adversarial objective; however, instead of nuisance estimation and refinement, it applies a lightweight control domain head directly on backbone features. This design ensures that improvements can be attributed to nuisance-aware refinement rather than changes in optimization signals.

Table VII summarizes the ablation results. NARR improves cross-dataset ranking performance substantially on DFDC (AUC 0.7052 vs. 0.6548) and also improves robustness under aggressive JPEG compression (JPEG30 AUC 0.8533 vs. 0.8467). Gains in-domain on FF++ are smaller (AUC 0.9378 vs. 0.9331), which is consistent with the intended role of NARR as a robustness plug-in rather than an in-domain accuracy booster.

TABLE VII
ABLATION: NARR VS. BASELINE (NO NARR)

Model	FF++ AUC	JPEG30 AUC	DFDC AUC	Celeb-DF AUC
Baseline (no NARR)	0.9331	0.8467	0.6548	0.6837
NARR (Ours)	0.9378	0.8533	0.7052	0.7113

F. Compression Robustness

Robustness to post-processing artifacts is evaluated using JPEG quality levels $\{100, 90, 75, 50, 30\}$. Table VIII reports AUC across compression levels.

NARR shows strong robustness in high-to-moderate compression regimes and controlled degradation at severe compression. The model reaches AUC 0.9379 at JPEG100 and remains above 0.92 through JPEG75, then decreases to 0.8888 (JPEG50) and 0.8533 (JPEG30). This trend indicates effective nuisance suppression while also revealing a limitation under aggressive quality loss.

TABLE VIII
JPEG COMPRESSION ROBUSTNESS (AUC)

Model	100	90	75	50	30
NARR (Ours)	0.9379	0.9348	0.9214	0.8888	0.8533
Baseline (no NARR)	0.9331	0.9257	0.9147	0.8811	0.8467
RFFR	0.9300	0.9298	0.9181	0.9022	0.8818
CLIPping the Deception	0.8201	0.8214	0.7954	0.7727	0.7711
Unlocking CLIP	0.9356	0.9208	0.8967	0.8457	0.8246
RoGA	0.8445	0.8438	0.8422	0.8402	0.8351
Forensics Adapter	0.7876	0.7068	0.6572	0.5629	0.4867
D^3	0.8480	0.8503	0.8005	0.7726	0.7735
Frequency-Aware	0.6729	0.6686	0.6806	0.6826	0.6802
ForgeLens	0.5930	0.5875	0.5899	0.5972	0.6027
Xception Detector	0.8426	0.8382	0.8298	0.8117	0.7855

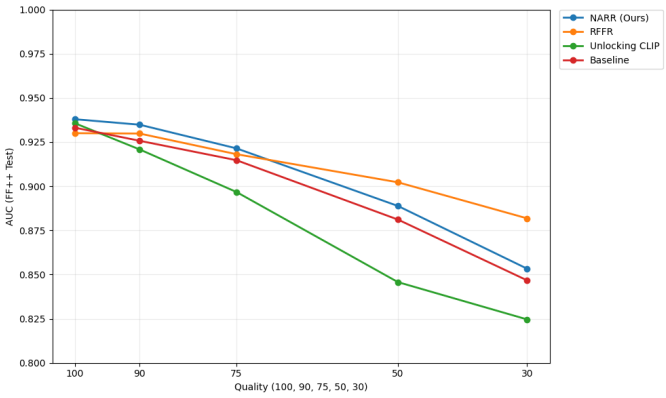

Fig. 3. Compression robustness comparison across benchmarked methods. NARR maintains high AUC under strong JPEG recompression, suggesting reduced reliance on compression-specific artifacts.

G. Cross-Dataset Tests

Cross-dataset generalization is assessed by evaluating models trained only on FF++ against DFDC and Celeb-DF without fine-tuning. NARR achieves AUC 0.7052 on DFDC and 0.7113 on Celeb-DF, showing stable transfer across two unseen datasets under the same lightweight training budget.

TABLE IX
CROSS-DATASET AUC

Model	DFDC AUC	Celeb-DF AUC
NARR (Ours)	0.7052	0.7113
Baseline (no NARR)	0.6548	0.6837
RFFR	0.6375	0.7835
CLIPping the Deception	0.5126	0.5896
Unlocking CLIP	0.5270	0.7344
RoGA	0.5398	0.7794
Forensics Adapter	0.5563	0.6136
D^3	0.5152	0.6425
Frequency-Aware	0.5130	0.5442
ForgeLens	0.5343	0.5602
Xception Detector	0.4198	0.7013

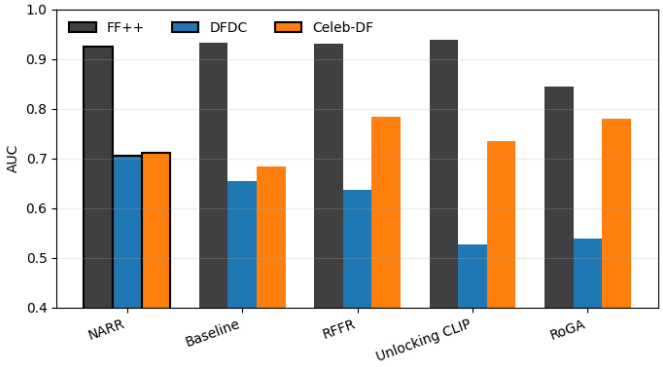

Fig. 4. Cross-dataset generalization across benchmarked methods. Grouped bars illustrate AUC differences between in-domain (FF++) and cross-dataset settings (DFDC, Celeb-DF) under identical training conditions.

VI. DISCUSSION

Results show that NARR provides modest improvements on in-domain FF++ evaluation but yields larger gains under distribution shifts, including strong JPEG compression and cross-dataset transfer. This behavior reflects the design goal of NARR: reducing reliance on unstable nuisance signals rather than increasing in-domain discriminative power.

In the baseline detector, semantic and nuisance cues are entangled within shared backbone features. Although invariance and domain-adversarial objectives encourage robustness, they operate on the same representation space and may still allow shortcut learning. NARR instead introduces an explicit nuisance estimation stage prior to tokenization, enabling targeted suppression of instability patterns through dual-channel gating. Applying refinement before the transformer encoder prevents nuisance amplification during global attention, which likely contributes to improved transfer performance.

The computational overhead remains minimal, with only a 1.9% increase in FLOPs relative to the baseline. This suggests that robustness gains arise from representation refinement rather than increased model capacity. However, performance still degrades under extreme compression, indicating that severe information loss remains a challenge. Additionally, experiments are limited to frame-level inputs and short training schedules, leaving temporal modeling and extended optimization for future work.

VII. CONCLUSION

We introduced NARR, a nuisance-aware representation refinement module designed as a lightweight plug-in for CNN-based deepfake detectors. By explicitly estimating and suppressing nuisance signals before tokenization, NARR improves robustness to compression artifacts and cross-dataset shifts while maintaining strong in-domain performance.

Under a unified benchmarking protocol, NARR achieves consistent gains in cross-dataset AUC with only a 1.9% increase in computational cost, demonstrating that targeted refinement can improve generalization without scaling model capacity. Future work will explore integration with temporal modeling and broader applications of nuisance-aware refinement in robustness-critical vision tasks.