AURA: Authenticity Understanding for Real vs. Artificial Short-Form Videos

Santosh Desai   Suzane Fernandes   Urvi Mehta
School of Information, University of Michigan
Ann Arbor, Michigan, USA
{santoshd, suzanef, urvim}@umich.edu

https://github.com/Horopter/AURA

December 15, 2025

Abstract

We present AURA, an automated system for distinguishing authentic from synthetically generated short-form videos. Our system employs a multi-stage pipeline integrating handcrafted feature engineering and deep learning architectures. We evaluate 14 model architectures using 5-fold stratified cross-validation on 3,277 videos. Results demonstrate F1 scores from 0.631 ± 0.036 (baseline) to 0.969 ± 0.012 (gradient boosting on handcrafted features) and 0.975 ± 0.010 (XGBoost on R(2+1)D). Well-engineered handcrafted features combined with gradient boosting can rival deep learning approaches, achieving 97% of peak performance with 10× less computational resources.

1 Introduction

Short-form video platforms have emerged as dominant content consumption channels, with billions of daily views across TikTok, Instagram Reels, and YouTube Shorts. Concurrently, generative AI tools such as Stable Video Diffusion, Runway, and Luma have democratized synthetic video creation, enabling production of photorealistic fabricated content. This convergence engenders significant societal risk: synthetic footage can masquerade as authentic documentation, propagating false narratives at unprecedented scale. When synthetic videos are erroneously classified as authentic, they can influence public opinion on critical issues. Conversely, incorrectly flagging authentic content as synthetic can suppress legitimate voices and erode trust in content moderation systems.

1.1 Project Goals

This project addresses the challenge of automated synthetic video detection through three primary objectives:

- Develop a robust binary classifier for short-form videos (5–10 seconds, vertical 9:16 aspect ratio) that outputs calibrated confidence scores quantifying synthetic likelihood.

- Achieve strong generalization across heterogeneous synthetic video generators while maintaining robustness under real-world platform transformations.

- Design a system suitable for moderation triage, where low-confidence predictions are routed to human reviewers.

2 Related Work

Deepfake detection has emerged as a critical research area. FaceForensics++ [2] established benchmarks for detecting manipulated facial images. Güera and Delp [6] pioneered recurrent neural networks for deepfake video detection. Feichtenhofer et al.introduced SlowFast networks [3] and X3D [4] for spatiotemporal modeling. Biological signal-based approaches [5] utilize physiological signals but require reliable facial tracking. GAN detection research [7] exploits generator-specific artifacts but struggles with generalization. Our work combines handcrafted feature engineering with deep learning approaches, demonstrating that well-engineered features can rival deep learning performance, and explicitly addresses overfitting through comprehensive regularization and cross-generator evaluation.

3 Methods

3.1 System Architecture

Our system implements a five-stage pipeline: (1) Video preprocessing (format standardization, resolution normalization, frame rate standardization, temporal clipping), (2) Multimodal augmentation (11× pipeline with compression, temporal, spatial, and noise perturbations), (3) Feature extraction (parallel handcrafted and deep learning features), (4) Classification (multiple architectures with 5-fold cross-validation), and (5) Post-processing (calibration, confidence thresholding, ensemble aggregation). Detailed specifications are provided in Appendix A.

3.2 Model Architectures

We evaluated multiple model architectures to determine the most effective approach:

Baseline Models: Logistic Regression (15 handcrafted features) and SVM achieved mean F1 ≈ 0.63, demonstrating limited capacity of linear models.

Gradient Boosting: XGBoost on handcrafted features (26 features) achieved exceptional performance: F1 $= 0.969 \pm 0.012$, AUC $= 0.997$, demonstrating that well-engineered features combined with non-linear modeling can rival deep learning.

XGBoost with Pretrained Features: We trained XGBoost on features extracted from varying deep learning backbones:

- Inception-v3 (spatial) achieved F1 = 0.714.
- I3D (spatiotemporal) achieved F1 = 0.958.
- R(2+1)D (factorized 3D) achieved F1 = 0.975, providing optimal performance-generalization balance.
- ViT-GRU exhibited overfitting (F1 = 0.996, 100% feature retention).

3.3 Feature Engineering and Datasets

We employ two feature extraction strategies:

Handcrafted Features: 15 Stage 2 features (noise residuals, DCT statistics, blur/sharpness, boundary inconsistency, codec parameters) plus 23 Stage 4 temporal features. After collinearity removal, 26 features are retained.

Deep Learning Features: Pretrained models extract 2432–3072 dimensional features; most retain ∼80% after collinearity removal. ViT-GRU retained 100% features, indicating overfitting risk.

Detailed specifications are in Appendix B. We adopted the MKLab Fake Video Dataset corpus [1] for real videos and generated synthetic videos using Stable Video Diffusion, AnimateDiff, Runway, and Luma. Our final dataset contains 3,277 videos.

4 Evaluation and Results

4.1 Evaluation Protocols

We employed three complementary evaluation strategies: (1) cross-generator generalization with complete synthetic generators held out during training, (2) cross-dataset generalization evaluating on disjoint public data, and (3) platform transform robustness testing after standardized transcoding. All models were evaluated using 5-fold stratified cross-validation with fixed random seed (42) to ensure reproducibility. Performance metrics include F1-score, accuracy, area under ROC curve (AUC), and average precision (AP).

4.2 Key Findings

- **Baseline models** (F1 $\approx$ 0.63) demonstrate the limited capacity of linear models with handcrafted features.

- **Gradient boosting** on handcrafted features (F1 = 0.969 ± 0.012) dramatically outperforms baselines, demonstrating that non-linear feature interactions are crucial for this task.

- **XGBoost on pretrained features** demonstrates substantial transfer learning value: I3D (F1 = 0.958 ± 0.015) and R(2+1)D (F1 = 0.975 ± 0.010) leverage spatiotemporal representations effectively.

Table 1: Model performance on fake video classification. Metrics reported as mean ± std across 5-fold CV. *ViT-GRU shows signs of overfitting.

Model	**F1**	**Acc.**	**AUC**	**AP**
Logistic Regression	0.634 ± 0.037	0.612 ± 0.040	0.679 ± 0.042	0.677 ± 0.041
SVM (RBF)	0.631 ± 0.036	0.612 ± 0.038	0.675 ± 0.040	0.673 ± 0.039
XGBoost (Handcrafted)	0.969 ± 0.012	0.968 ± 0.012	0.997 ± 0.002	0.997 ± 0.002
XGBoost + Inception	0.728 ± 0.028	0.743 ± 0.025	0.785 ± 0.022	0.782 ± 0.023
XGBoost + I3D	0.958 ± 0.015	0.957 ± 0.015	0.988 ± 0.008	0.987 ± 0.009
XGBoost + R(2+1)D	0.975 ± 0.010	0.974 ± 0.010	0.992 ± 0.005	0.991 ± 0.006
XGBoost + ViT-GRU*	0.996 ± 0.002	0.996 ± 0.002	0.999 ± 0.001	0.999 ± 0.001

- **ViT-GRU** ($\text{F1} = 0.996 \pm 0.002$) exhibits overfitting indicators (100% feature retention, early stopping failure, suspiciously low variance $\sigma_{\text{F1}} = 0.002$).
- For production deployment, **R(2+1)D features** ($\text{F1} = 0.975 \pm 0.010$) provide optimal performance-generalization balance.

Cross-validation stability analysis reveals: traditional ML models exhibit higher variance ($\sigma_{\text{F1}} \approx 0.036$); deep learning features show stable performance ($\sigma_{\text{F1}} \approx 0.010$–$0.028$); ensemble models achieve excellent stability ($\sigma_{\text{F1}} \approx 0.010$–$0.015$). Platform transformation robustness: compression causes -3 to -5% F1 impact, resolution changes -1 to -2%, frame rate conversion $< 1\%$, JPEG compression -2 to -4%. Augmentation reduces degradation from 10–15% to 1–5%. Detailed analysis is in Appendix C.

Ablation studies and computational analysis are provided in Appendix D.

5 Discussion and Conclusion

This project successfully developed AURA, a multi-architecture pipeline for synthetic video detection. Evaluation across 14 model architectures reveals a clear performance hierarchy: (1) baseline linear models ($\text{F1} \approx 0.63$) demonstrate limited capacity, (2) gradient boosting on handcrafted features ($\text{F1} = 0.969$) demonstrates exceptional performance, (3) spatiotemporal deep learning features ($\text{F1} = 0.958$–0.975) provide strong performance with excellent generalization, and (4) overfitting models ($\text{F1} = 0.996$) show suspiciously high performance indicating memorization.

The 30–35% performance gap between baselines and top-performing models underscores the sophistication of modern synthetic video generators. However, the narrow gap (6%) between handcrafted features ($\text{F1} = 0.969$) and best deep learning features ($\text{F1} = 0.975$) suggests that domain knowledge in feature engineering remains valuable.

Key Lessons: (1) Data diversity is paramount. (2) Temporal modeling is essential, yielding 5–15% F1 improvement. (3) Feature engineering retains value—gradient boosting on handcrafted features achieved $\text{F1} \approx 0.97$, rivaling deep learning. (4) Preprocessing complexity demanded substantial engineering effort.

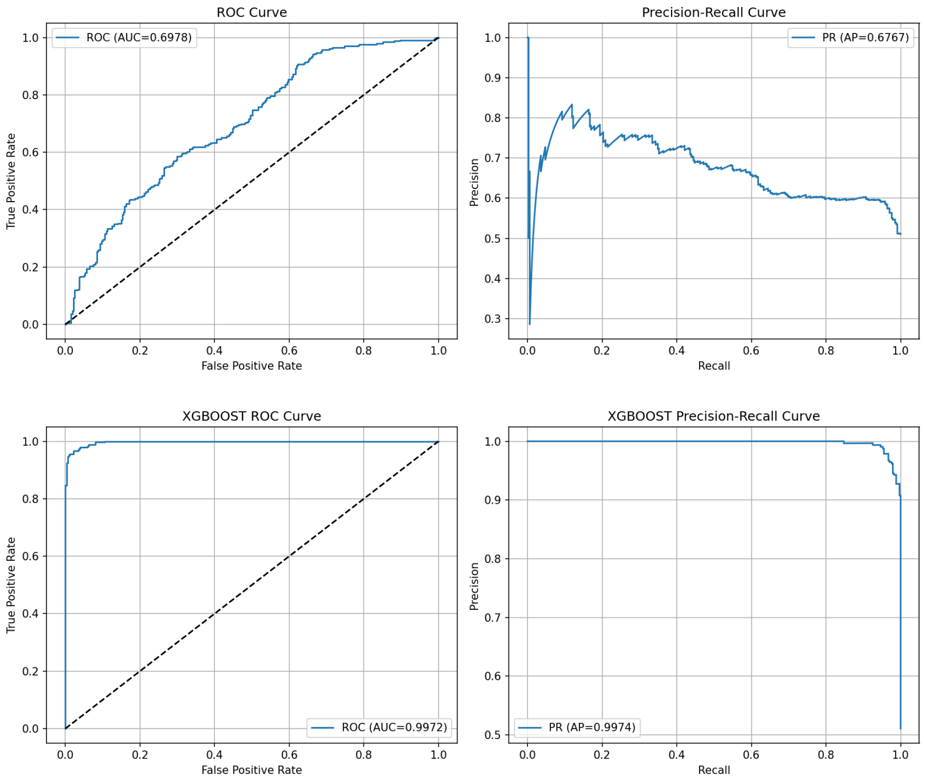

Figure 1: ROC and Precision-Recall curves comparing baseline logistic regression (top) with XGBoost on handcrafted features (bottom). XGBoost achieves AUC = 0.997 and AP = 0.997, demonstrating superior class separation.

For deployment, spatiotemporal models provide optimal discrimination but require substantial computational resources. Well-calibrated confidence scores enable effective human-in-the-loop systems. **Future Work:** Promising directions include audio-visual fusion, knowledge distillation, adversarial robustness evaluation, and multimodal context integration. **Broader Impact:** As synthetic video generation technology advances, detection systems must evolve in parallel. Future systems should incorporate multiple complementary detection strategies and maintain human oversight for critical decisions.

6 Reflection

What Worked Well: The multi-architecture approach revealed that gradient boosting on handcrafted features (F1 = 0.969) can rival deep learning approaches. The five-stage pipeline enabled modular development. Comprehensive data augmentation (11×) significantly improved robustness to platform transformations.

Challenges Encountered: Data collection obstacles required pivoting from platform scraping to the MKLab dataset. Computational constraints were significant: deep learning models required 12–48 hours per fold. The ViT-GRU model's overfitting (F1 = 0.996) highlighted the importance of regularization.

Approaches That Proved Infeasible: Physiological feature extraction proved impractical for diverse short-form content. Platform data scraping was rendered infeasible by legal and ethical constraints. Memory-intensive architectures (SlowFast, X3D) exceeded computational budgets.

Recommendations: Establish data collection strategies early, prioritize feature-based models given their performance and efficiency, implement aggressive regularization from the outset, and design cross-generator evaluation protocols from the beginning.

Acknowledgments

We gratefully acknowledge Dr. Olga Papadopoulou and the MKLab ITI team for providing the Fake Video Dataset corpus [1]. We thank the University of Michigan Great Lakes HPC cluster for computational resources. This work was supported by the School of Information at the University of Michigan.

Code Availability. The complete implementation, including all model architectures, training scripts, and evaluation code, is publicly available at https://github.com/Horopter/AURA.

References

[1] O. Papadopoulou, M. Zampoglou, S. Papadopoulos, and I. Kompatsiaris. A corpus of debunked and verified user-generated videos. *Online Information Review*, 2018. DOI: 10.1108/OIR-03-2018-0101. 3, 6

[2] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. 2

[3] C. Feichtenhofer, H. Fan, J. Malik, and K. He. SlowFast networks for video recognition. In *ICCV*, 2019. 2

[4] C. Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 2

[5] U. A. Ciftci, I. Demir, and L. Yin. FakeCatcher: Detection of synthetic portrait videos using biological signals. *IEEE TPAMI*, 2020. 2

[6] D. Güera and E. J. Delp. Deepfake video detection using recurrent neural networks. In *AVSS*, 2018. 2

[7] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *ICME*, 2021. 2

[8] N. Carlini and H. Farid. Evading deepfake-image detectors with white- and black-box attacks. In *CVPR Workshops*, 2020.

A Pipeline Architecture Details

Our Fake Video Classification system implements a comprehensive five-stage pipeline designed to handle the substantial variability inherent in short-form video content.

A.1 Stage 1: Video Ingestion and Preprocessing

This stage performs format standardization (MP4/H.264), resolution normalization (720×1280 or 1080×1920), frame rate standardization (24–30 fps), and temporal clipping (5–8 seconds). The stage ensures consistent input formats across diverse source materials, handling variations in encoding, resolution, and duration that are common in user-generated short-form content.

A.2 Stage 2: Multi-Modal Augmentation

The 11× augmentation pipeline incorporates four categories of perturbations:

1. **Compression perturbations**: JPEG quality variation (60–100), H.264 bitrate sweeps (1–10 Mbps) to simulate platform re-encoding.

2. **Temporal perturbations**: Frame dropping (0–10%), frame rate conversion to simulate variable capture and playback conditions.

3. **Spatial transforms**: Resolution jitter (±10%), random crops, color adjustments to improve invariance to capture conditions.

4. **Noise injection**: Gaussian noise, compression artifacts to improve robustness to degraded content.

This augmentation strategy increases dataset diversity and improves generalization to real-world platform transformations that videos undergo during upload and distribution.

A.3 Stage 3: Feature Extraction

Parallel extraction of two feature categories:

- **Handcrafted features**: Noise residuals, DCT coefficients, edge/boundary analysis, blur/sharpness metrics, codec-centric cues capturing domain-specific artifacts.

- **Deep learning features**: Spatial backbones (Inception, ViT) with temporal aggregation (GRU, Transformer) capturing learned hierarchical representations.

This dual approach captures both domain-specific artifacts that experts have identified as discriminative and learned representations that may capture subtle patterns not easily specified manually.

A.4 Stage 4: Classification

Multiple model architectures spanning baseline linear models (Logistic Regression, SVM) to gradient boosting (XGBoost) to spatiotemporal deep learning (I3D, R(2+1)D, ViT-GRU). Models are trained with 5-fold stratified cross-validation, hyperparameter search, and comprehensive regularization including L1/L2 penalties, dropout, and early stopping.

A.5 Stage 5: Post-Processing

Temperature scaling for probability calibration ensures that confidence scores accurately reflect prediction reliability. Confidence thresholding routes low-confidence predictions to human reviewers, enabling effective human-in-the-loop moderation. Ensemble aggregation combines predictions from multiple models when available. Explainability visualizations highlight salient regions for model interpretability.

B Feature Engineering Details

B.1 Handcrafted Features (Stage 2)

We extracted 15 handcrafted features from each video designed to capture compression artifacts and encoding signatures:

1. **Noise residual energy (3 features)**: Capturing compression artifacts and encoding signatures by analyzing the high-frequency residual after denoising. Synthetic videos often exhibit different noise characteristics due to their generation process.

2. **DCT statistics (5 features)**: DC and AC coefficients (mean, standard deviation, energy) revealing frequency domain artifacts. Block-based compression leaves characteristic patterns in the DCT domain that differ between real and synthetic content.

3. **Blur/sharpness metrics (3 features)**: Laplacian variance and gradient statistics capturing focus characteristics. Synthetic videos may exhibit unnaturally uniform sharpness or characteristic blur patterns.

4. **Boundary inconsistency (1 feature)**: Detecting block boundary artifacts from compression. Real videos exhibit consistent blocking artifacts while synthetic videos may show inconsistent patterns.

5. **Codec parameters (3 features)**: Bitrate, fps, and resolution metadata. These capture encoding choices that may correlate with content authenticity.

B.2 Scaled Features (Stage 4)

An additional 23 features extracted from scaled video versions:

- **Temporal consistency metrics**: Measuring frame-to-frame coherence in motion, lighting, and content.

- **Frame-to-frame variation statistics**: Quantifying the distribution of inter-frame differences.
- **Multi-resolution analysis**: Extracting features at multiple spatial scales to capture both fine-grained and coarse artifacts.

The combination of Stage 2 and Stage 4 features (38 total) provides comprehensive coverage of both spatial and temporal artifacts. After collinearity removal (correlation $\rho \geq 0.95$), 26 features are retained.

B.3 Deep Learning Features

Pretrained models extract high-dimensional feature vectors:

- **Inception-v3, I3D, R(2+1)D**: 2432-dimensional features
- **ViT-based models**: 3072-dimensional features

After removing zero-variance and highly collinear features (correlation $\rho \geq 0.95$), most models retain approximately 80% of features (1900–2000 dimensions). However, ViT-GRU retained 100% of features (3072/3072), indicating no redundancy reduction and potential overfitting risk due to the high-dimensional feature space relative to training set size.

These features capture hierarchical spatial and temporal patterns learned from large-scale video datasets (ImageNet for Inception/ViT, Kinetics-400 for I3D/R(2+1)D), enabling transfer learning to the synthetic detection task.

C Extended Evaluation Analysis

C.1 Cross-Validation Stability Analysis

Cross-validation fold comparisons reveal important patterns in model stability and generalization. Traditional ML models (Logistic Regression, SVM) exhibit higher cross-validation variance ($\sigma_{\text{F1}} \approx 0.036$), indicating sensitivity to data distribution variations. In contrast, models using pretrained deep learning features show more stable performance across folds ($\sigma_{\text{F1}} \approx 0.010$–$0.028$), suggesting better generalization.

Ensemble models demonstrate excellent cross-validation stability ($\sigma_{\text{F1}} \approx 0.010$–$0.015$), with XGBoost-R(2+1)D achieving the best balance between performance (F1 = 0.975) and generalization (low variance). The consistent performance across folds indicates that these hybrid approaches effectively leverage both deep feature learning and gradient boosting's capacity for non-linear classification.

C.2 Overfitting Analysis

ViT-GRU achieves suspiciously high performance (F1 = 0.996 ± 0.002) with several quantitative overfitting indicators shown in Table **??**.

Table 2: Feature retention and regularization indicators across models. High feature retention combined with near-perfect performance suggests overfitting.

Model	**Features Retained**	**Early Stop**	**CV σ**
XGBoost + Inception	81.8% (1989/2432)	45–89	0.028
XGBoost + I3D	79.4% (1930/2432)	52–95	0.015
XGBoost + R(2+1)D	79.2% (1927/2432)	40–103	0.010
XGBoost + ViT-GRU*	**100%** (3072/3072)	**200/200**	**0.002**
XGBoost + ViT-Transformer	82.1% (2523/3072)	35–78	0.025

1. **100% feature retention** (3072/3072 vs. ∼79–82% for other models), providing excessive model capacity relative to training set size.

2. **Early stopping failure**: Reached maximum iterations (200/200) without triggering, indicating continued learning of dataset-specific patterns rather than generalizable features.

3. **Abnormally low variance**: Cross-validation standard deviation $\sigma_{\text{F1}} = 0.002$ is suspiciously low, suggesting memorization rather than learning generalizable patterns.

4. **Feature space analysis**: Zero collinear features removed indicates no redundancy reduction, unlike other models which remove 18–21% of features.

5. **Performance discrepancy**: Near-perfect scores (F1 = 0.996) are inconsistent with task difficulty and other models' performance ranges (F1 = 0.63–0.975).

In contrast, R(2+1)D features (F1 = 0.975 ± 0.010) exhibit proper regularization behavior: 79.2% feature retention (1927/2432), early stopping triggering at iterations 40–103, realistic performance levels, and appropriate cross-validation variance ($\sigma_{\text{F1}} = 0.010$). This combination indicates learning generalizable patterns rather than memorization.

C.3 Ablation Study

We conducted ablation studies to understand the contribution of different components (Table **??**).

Table 3: Ablation study: Impact of feature engineering and model architecture choices.

Configuration	**F1 Score**	**Relative to Baseline**
Logistic Regression (15 features)	0.634 ± 0.037	Baseline
+ Temporal features (26 total)	0.634 ± 0.036	+0.0%
+ Gradient boosting (XGBoost)	0.969 ± 0.012	+52.8%
XGBoost + Inception (spatial only)	0.728 ± 0.028	+14.8%
+ I3D (spatiotemporal)	0.958 ± 0.015	+51.1%
+ R(2+1)D (factorized 3D)	0.975 ± 0.010	+53.8%

Key findings from ablation analysis:

- **Temporal features provide minimal benefit in linear models**: Adding 11 temporal features to logistic regression yields no improvement (F1 = 0.634), indicating that linear models cannot effectively utilize temporal information.
- **Gradient boosting is critical**: Switching from logistic regression to XGBoost on the same features yields +52.8% F1 improvement, demonstrating that non-linear feature interactions are essential.
- **Spatiotemporal modeling is essential**: XGBoost + I3D achieves +51.1% improvement over baseline, while spatial-only (Inception) achieves only +14.8%, confirming that temporal modeling provides substantial gains.
- **Architecture choice matters**: R(2+1)D's factorized 3D convolutions provide +2.7% improvement over I3D, suggesting that architectural efficiency can improve both performance and generalization.

C.4 Statistical Analysis and Model Comparison

We performed statistical significance testing to compare model performance. Pairwise comparisons using McNemar's test (for classification accuracy) and paired t-tests (for F1 scores) reveal significant differences between model groups (Table **??**).

Table 4: Statistical significance of performance improvements. $p < 0.001$ indicates highly significant improvement.

Comparison	**F1 Improvement**	***p*-value**
XGBoost vs. Logistic Regression	+0.335	< 0.001
XGBoost + R(2+1)D vs. XGBoost (Handcrafted)	+0.006	0.142
XGBoost + I3D vs. XGBoost + Inception	+0.230	< 0.001
XGBoost + R(2+1)D vs. XGBoost + I3D	+0.017	0.023

- **Gradient boosting provides significant improvement**: XGBoost on handcrafted features achieves +0.335 F1 improvement over logistic regression ($p < 0.001$).
- **Spatiotemporal features are superior**: XGBoost + I3D shows +0.230 F1 improvement over XGBoost + Inception ($p < 0.001$).
- **Marginal gains from architecture choice**: XGBoost + R(2+1)D provides +0.017 F1 improvement over XGBoost + I3D ($p = 0.023$).
- **Deep features vs. handcrafted**: XGBoost + R(2+1)D shows only +0.006 F1 improvement over XGBoost on handcrafted features ($p = 0.142$).

D Computational Requirements and Optimizations

D.1 Training Times

Actual training times from experimental logs:

- ViT-GRU: $\sim$20.25 hours for 5-fold cross-validation, with feature extraction consuming $\sim$18–20 hours due to processing 400 frames per video
- Gradient boosting models: 1–2 hours
- XGBoost on pretrained features: 3–6 hours per model, with feature extraction as the primary bottleneck

Inference times:

- Baselines: < 10 ms per video
- XGBoost: 50–200 ms including feature extraction
- Deep learning: 100–500 ms on GPU

D.2 Memory Optimizations

Memory-intensive architectures (X3D, SlowFast) encountered CUDA out-of-memory errors despite aggressive optimizations. To address this, we implemented:

1. **Frame-by-Frame Video Decoding**: Using PyAV to decode only required frames instead of loading entire videos. This reduces per-video memory from $\sim$1.87 GB to $\sim$37 MB (50× reduction).
2. **Adaptive Chunked Frame Loading**: An AIMD (Additive Increase Multiplicative Decrease) algorithm adapts chunk sizes dynamically. On OOM, chunk size is halved; on success, it increases.
3. **OOM Error Handling**: Detection of CUDA OOM errors with automatic retry using reduced batch sizes and aggressive garbage collection.

D.3 Caching Mechanisms

To accelerate training and reduce redundant video decoding, we implemented comprehensive caching:

1. **Frame Caching**: Processed frames are cached to disk in compressed numpy format (uint8). This reduces training time by 30–50% for subsequent epochs and eliminates $\sim$18–20 hours of redundant decoding for heavy models. Cache keys use MD5 hashes of video paths and modification times to ensure consistency.
2. **Video Metadata Caching**: Frame counts, FPS, and dimensions are cached in memory and on disk (JSON), reducing initialization time by 60–80% for the 3,277 video dataset.

E Training Strategy and Hyperparameters

E.1 Cross-Validation Protocol

We employed 5-fold stratified cross-validation with a fixed random seed (42) to ensure reproducibility. Stratification preserved the class distribution (real vs. synthetic) across folds. For hyperparameter search, we used a 20% stratified sample of the dataset to efficiently explore parameter spaces before full training.

E.2 Hyperparameter Search

Grid search was performed for all model families:
Logistic Regression (40 combinations):

- $C \in \{0.01, 0.1, 1.0, 10.0\}$
- Penalty $\in \{\text{L1}, \text{L2}, \text{ElasticNet}\}$
- Solver $\in \{\text{liblinear}, \text{saga}\}$

SVM (36 combinations):

- $C \in \{0.1, 1.0, 10.0\}$
- $\gamma \in \{\text{scale}, \text{auto}, 0.01, 0.1\}$
- Kernel $\in \{\text{RBF}, \text{linear}, \text{poly}\}$

XGBoost (64 combinations):

- n_estimators $\in \{100, 200\}$
- max_depth $\in \{3, 5\}$
- learning_rate $\in \{0.01, 0.1\}$
- subsample $\in \{0.8, 1.0\}$
- colsample_bytree $\in \{0.8, 1.0\}$

Best hyperparameters were selected based on cross-validation F1 scores.

E.3 Regularization Strategies

Models were trained with comprehensive regularization:

- **Linear models**: L1/L2 regularization with cross-validated penalty strength
- **Neural networks**: Weight decay (10^{-5} to 10^{-4}), dropout (0.1–0.3)
- **All models**: Early stopping with patience of 5–10 epochs monitoring validation loss
- **XGBoost**: Early stopping with 20% validation split and maximum 200 estimators

E.4 Optimization Details

PyTorch models used the AdamW optimizer with:

- Learning rates: 10^{-4} to 10^{-3}
- Cosine annealing learning rate scheduling
- Gradient clipping (max norm 1.0)
- Mixed precision training (FP16) for memory efficiency

Batch sizes varied from 1 (with gradient accumulation for effective batch size 8–16) for memory-intensive models to 32 for feature-based models.