# FitCoachAR: Real-Time Adaptive Exercise Coaching via Pose Estimation and AR Feedback

Yangyang Zhang    Maximilian Fuchs    Seyedmohamad Mirhoseininejad

CAS 772: Mobile Data Analytics

## 1 Introduction

Home and gym fitness applications increasingly employ AI-based pose estimation to count exercise repetitions, yet most still lack real-time, adaptive coaching on movement quality. Early research systems such as Pose Trainer [3] and AIFit [5] demonstrate that 3D pose analysis can detect posture errors and evaluate performance, but their pipelines remain offline, relying on recorded videos and computationally heavy 3D reconstruction. They provide limited personalization and cannot adapt feedback dynamically during a workout. Recent work such as ARFit [7] integrates augmented-reality visualization to guide users through motion sequences, but still applies generic thresholds and lacks quantitative evaluation of performance consistency.

Commercial applications (e.g., Top Pushup [1], QuickPose [8]) have popularized real-time motion counting and form tracking using smartphone cameras. However, they mainly focus on rep detection and simple form classification without deeper analysis—such as distinguishing good vs. bad repetitions, assessing tempo stability, or tracking per-joint improvement over time. Moreover, most commercial apps rely on fixed thresholds and provide limited adaptive or personalized feedback.

FitCoachAR aims to bridge the gap between academic models and consumer applications by providing a lightweight, real-time, and adaptive coaching system. It monitors exercises via 2D pose estimation, detects common form errors, and delivers feedback through augmented-reality overlays and coach-style natural language cues. The system personalizes its critique level through a short calibration phase and produces both live corrections and post-session summaries.

This project addresses three major motivations:

- **Accessibility**: Eliminate the dependency on motion-capture equipment and high-end GPUs.
- **Personalization**: Adapt thresholds and critique sensitivity ($\delta$) to each user.
- **Interactivity**: Transform static, after-exercise evaluation into continuous, real-time feedback that enhances motivation and engagement.

## 2 Related Work

### 2.1 Academic Systems

Pose Trainer (2020) [3] applied rule-based analysis of 2D skeletons from OpenPose [2] to classify push-ups and squats. While effective for offline evaluation, it operates on recorded videos and uses fixed thresholds, providing no real-time correction or personalization.

AIFit (CVPR 2021) [5] introduced the Fit3D motion-capture dataset and a 3D feedback model capable of joint-level error localization using a deviation parameter ($\eta$) to control strictness. However, it depends on full-sequence 3D reconstruction (MubyNet) and produces template-based text feedback, making it computationally expensive and less adaptive.

ARFit (IMWUT 2023) [7] added mobile AR overlays for pose visualization, showing that spatial feedback improves exercise learning. Yet, its thresholds remain generic and it lacks quantitative analytics such as repetition consistency or tempo tracking, focusing mainly on visual guidance.

### 2.2 Commercial Applications

Apps like Top Pushup [1] and QuickPose [8] popularized real-time repetition counting using smartphone cameras. Their analysis remains binary—labeling repetitions as correct or incorrect—without distinguishing motion quality, tempo stability, or per-joint improvements. They also rely on static thresholds and simple chatbot-style feedback rather than adaptive, natural-language coaching.

### 2.3 Gap Summary

Across both research and consumer systems, key limitations persist:

- Offline or delayed feedback (Pose Trainer, AIFit).
- Lack of personalization (global thresholds or $\delta$ not user-specific).
- Shallow analysis (no holistic quality metrics).
- Limited interactivity (no context-aware language feedback).

FitCoachAR addresses these gaps through:

- Real-time 2D tracking and online calibration.
- Adaptive deviation sensitivity ($\eta$) for user-specific tolerance.
- AR-based spatial feedback.
- LLM-driven natural coaching for motivating, context-aware guidance.

This integration combines AIFit's interpretability with ARFit's usability, achieving real-time, personalized exercise feedback on consumer hardware.

## 3 Methodology

### 3.1 Pose Acquisition

We employ MediaPipe Pose [6] for real-time keypoint extraction (33 2D joints). The stream is smoothed using a One-Euro filter to reduce jitter and ensure sub-100 ms latency.

### 3.2 Online Repetition Segmentation

Following AIFit's unsupervised approach, motion periodicity is analyzed from key joint angles (elbow, knee, hip). A simplified online state machine detects phase transitions—descent, bottom, ascent—using derivative sign changes with hysteresis. Each full cycle is labeled as one repetition.

## 3.3 Feature Computation and Calibration

From each repetition we compute angular features:

- Active joints: elbows, knees, shoulders (max, min, correlation).
- Passive joints: spine and pelvis (mean, standard deviation).

During calibration, each key joint or motion phase $i$ is analyzed over three "best-form" repetitions: The mean joint angle is

$$\bar{U}_i = \frac{1}{3}\sum_{r=1}^{3}\theta_i^{(r)} \tag{1}$$

Its deviation percentage from a canonical target $S_i$ is

$$\eta_i = \frac{\bar{U}_i - S_i}{S_i} \tag{2}$$

The pair $(\bar{U}_i, \eta_i)$ forms the personalized baseline and offset for that user. During runtime, for each joint or phase, the system:

1. Measures the current angle $\theta_i$
2. Computes percentage deviation $e_i = \frac{|\theta_i - \bar{U}_i|}{|\bar{U}_i|}$

A manually set critic parameter $\delta$ determines grading bands:

- Good: $e_i < \delta$
- Relatively good: $\delta \leq e_i < 1.2\delta$
- Bad: $e_i \geq 1.2\delta$

Repetition-level and session-level scores aggregate these joint/phase grades to summarize overall performance.

## 3.4 AR Visualization

A transparent overlay renders skeleton lines, target "shadow" poses, arrows toward ideal joint positions, and colored angle sectors (green = within band, red = error). Implementation uses HTML Canvas (web); later versions can migrate to Unity + AR Foundation for full 3D anchoring.

## 3.5 FormScript: Interpretable Feedback via FormCodes

To provide meaningful, explanatory feedback, we implemented "FormScript", a rule-based system inspired by the "PoseScript" methodology [4]. While deep learning models can classify good vs. bad form, they often lack interpretability. FormScript solves this by discretizing continuous kinematic concepts into human-readable categories (FormCodes) and then applying logical rules (Super FormCodes) to generate feedback.

### 3.5.1 FormCode Categorization

Following the taxonomy defined in PoseScript, we implemented low-level logic units called "FormCodes". We extended the original static definitions (Angles, Distances) to include dynamic temporal properties required for fitness analysis.

**Spatial Relations (Static PoseCodes)** These FormCodes analyze the body geometry at key moments (e.g., the bottom of a squat). They are directly adapted from the PoseScript paper's categories:

- **Angle Codes**: Measure the flexion of joints.
  - `squat_depth` (Squat): Categorizes the knee angle into "deep" ($< 90°$), "parallel" ($90° - 110°$), or "shallow" ($> 110°$).
  - `peak_flexion` (Bicep Curl): Categorizes elbow angle at top range into "full contraction", "partial", or "incomplete".
  - `wrist_angle` (Bicep Curl): Detects if wrist is "neutral" vs "flexed/extended".
- **Pitch & Roll Codes**: Measure the orientation of body parts relative to the vertical axis.
  - `torso_angle` (Squat): Categorizes trunk lean into "upright" ($< 20°$), "leaning", or "bent over".
- **Distance & Position Codes**: Measure relative positions between joints.
  - `stance_width` (Squat): Categorizes foot distance relative to shoulders ("narrow", "shoulder width", "wide").
  - `knee_forward_travel` (Squat): Checks z-axis position of knees relative to toes ("over toes", "behind toes").
  - `elbow_position` (Bicep Curl): Checks x/z-axis drift of elbow relative to torso ("anchored", "drift").

**Temporal Dynamics (Dynamic FormCodes)** We extended the PoseScript framework to analyze motion quality over time, which is critical for exercise:

- **Velocity Codes**: Measure the speed of movement phases.
  - `descent_speed` / `ascent_speed` (Squat): "controlled", "fast", "dive", "explosive".
  - `lift_speed` / `lower_speed` (Bicep Curl): "controlled", "slow", "fast drop".
- **Stability & Deviation Codes**: Measure variance or jerk (shakiness) during movement.
  - `knee_stability` (Squat): Deviation of knees from the feet-hip line ("stable", "wobble").
  - `movement_smoothness` (Squat): Jerk metric for overall control.
  - `swing_momentum` (Bicep Curl): Torso movement indicating cheating ("no swing", "body swing").
  - `hip_shift` (Squat): Symmetry metric for hip levelness.

### 3.5.2 Super FormCodes: The Rule Engine

To generate high-level coaching feedback, we aggregate these elementary FormCodes using logical production rules, termed "Super FormCodes". This allows us to define complex errors as combinations of simple states.

Figure 1: Logic tree showing how Super FormCodes are derived from elementary FormCodes.

Examples of Production Rules:

- **IF** `squat_depth` IS “deep” **AND** `knee_stability` IS “stable” **THEN** → `Good Rep`
- **IF** `torso_angle` IS “bent_over” **AND** `descent_speed` IS “dive” **THEN** → `Uncontrolled Forward Lean`
- **IF** `swing_momentum` IS “body_swing” **THEN** → `Cheating Form` (User is using momentum)

This hierarchical approach ensures that the system works like a human coach: first observing specific details (angles, speeds), then synthesizing them into a coherent critique.

### 3.6 Real-Time Visual and Rule-Based Feedback

To maintain sub-100 ms latency during exercise, real-time feedback uses deterministic rule-based templates combined with AR visualization:

- Template-based corrections: pre-defined feedback phrases mapped to specific error conditions (e.g., “Lower your hips” when pelvis deviation exceeds a threshold).
- AR overlay: skeleton lines, target “shadow” poses, arrows toward ideal joint positions, and colored angle sectors (green = within band, red = error).
- Audio cues (optional): brief beeps or spoken keywords for critical errors during high-intensity phases.

This approach ensures immediate, actionable feedback without network or computation delays, keeping users engaged in the flow state of exercise.

### 3.7 LLM-Driven Post-Session Analysis and Summary

After completing a workout session, accumulated performance data is processed by a Large Language Model to generate comprehensive, personalized coaching feedback.

**Input construction** Each session produces a detailed record such as:

```
{
  "exercise": "push_up",
  "total_reps": 15,
...
  "critic_level": 0.2
}
```

**LLM prompt design** A structured system prompt guides the model to act as a professional coach: “You are an expert fitness coach reviewing a completed workout session... Total response: 120–150 words.”

**Output example** “Excellent consistency on your 15 push-ups! Your tempo and range of motion were strong throughout...”

**Benefits of post-session LLM use**

- Richer context: full session history enables pattern detection.
- Biomechanical reasoning: LLM can explain why an error matters.
- Progression tracking: compare across sessions to show improvement.

- No latency constraints: 2–3 second generation is acceptable after workout.
- Cost-effective: one API call per session vs. hundreds during exercise.

Session statistics complement the LLM narrative with quantitative metrics: Total repetitions and success ratio, Per-joint error heatmap, Average tempo, range of motion, and phase durations, Personalized deviation parameter $\eta$ adjustments for next session.

# 4 Evaluation Plan

**Metrics and targets**

- Segmentation IoU: overlap of detected vs. manual rep boundaries; target $\geq 0.70$.
- Latency: end-to-end delay (camera → feedback); target $< 100$ ms.
- Error detection F1: accuracy of error detection vs. labeled data; target $> 0.80$.
- User study: 5 participants rate clarity and usefulness (1–5 Likert); target $\geq 4.0$ average.
- Personalization gain: precision improvement after calibration; target +10%.
- Summary usefulness: user rating of LLM feedback (1–5 Likert); target $\geq 4.2$ average.
- Summary relevance: expert validation of LLM feedback vs. video review; target $\geq 85\%$.

**Evaluation will use:**

- Fit3D dataset — for offline benchmarking of segmentation and feature thresholds.
- Self-recorded data — for calibration and real-time tests.

# 5 Expected Contributions

- Real-time, calibration-based extension of AIFit for interactive use.
- Lightweight AR feedback visualization framework requiring only a webcam.
- Integration of LLM-driven natural-language coaching with quantitative form analysis.
- Public demo and reproducible open-source code for academic use in mobile data analytics.

# References

[1] App Store. Top Pushup: AI push up counter. `https://apps.apple.com/us/app/top-pushup-ai-push-up-counter/id1644616673`, 2023. Accessed: 2025-12-10.

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Steven Chen and Richard R Yang. Pose Trainer: Correcting exercise posture using pose estimation. *arXiv preprint arXiv:2006.11718*, 2020.

[4] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. PoseScript: Linking 3D human poses and natural language. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2875–2889, 2024.

[5] Mihai Fieraru, Mihai Zanfir, Silviu-Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. AIFit: Automatic 3D human-interpretable feedback models for fitness training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9914–9923, 2021.

[6] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. MediaPipe: A framework for building perceptual pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

[7] Srdan Mandic, Ryan Tracy, and Misha Sra. ARFit: Pose-based exercise feedback with mobile AR. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction (SUI '23)*, page Article 45. ACM, 2023.

[8] QuickPose. QuickPose AI push up counter. `https://quickpose.ai/exercise-library/fitness/ai-push-up-counter`, 2024. Accessed: 2025-12-10.