From 3D Pose to Prose: Biomechanics-Grounded Vision–Language Coaching

Supplementary Material

This supplementary material provides the technical details referenced in the main paper. QEVD-bio-fit-coach dataset construction (Sec. 1), evaluation metrics including the biomechanics-aware LLM judge (Sec. 2), DoF annotation (Sec. 3), cycle detection (Sec. 5), implementation specifics (Sec. 6) and mathematical derivations (Sec. 7).

1. QEVD-bio-fit-coach Dataset Construction

This section details the construction pipeline for QEVD-bio-fit-coach, which augments the original QEVD-fit-coach [3] with biomechanically-grounded feedback annotations (referenced in the main paper Sec. 3.7).

1.1. Automated Generation Pipeline

The QEVD-bio-fit-coach dataset is created through an automated pipeline that applies the Motion Quality Context Module (Sec. 3.4.2 in the main paper) to transform colloquial coaching feedback into anatomically precise, quantified guidance.

Selective Feedback Replacement. The original QEVD-fit-coach dataset contains diverse feedback types: corrective instructions addressing form deviations (*e.g.*, "lower your body more", "keep your back straight"), motivational encouragement (*e.g.*, "good job!", "keep it up!"), and repetition counting (*e.g.*, "5 more reps").

We systematically rewrite *corrective and instructional feedback*, as these inherently describe biomechanical violations that can be quantified. At each corrective feedback timestamp, the trainer observed a form deviation; our pipeline detects and quantifies the actual biomechanical violation, then replaces the colloquial description with precise measurements. Motivational and counting feedback are retained unchanged, as they do not describe specific form issues.

Generation Procedure. For each video in the original QEVD-fit-coach dataset, we apply the complete BioCoach pipeline to generate biomechanical context:

1. Extract 3D skeletal kinematics using HSMR + SKEL (Sec. 6.3) and morphometric measurements (main paper Sec. 3.4.1)
2. Apply the trained DoF selection network $\mathcal{A}_\theta$ (Sec. 3) to identify exercise-specific salient joints $\mathcal{J}^*$
3. At each corrective/instructional feedback timestamp, run the Motion Quality Context Module pipeline (main paper Sec. 3.4.2): cycle detection, reference alignment, and biomechanical constraint evaluation to generate structured motion context $\mathcal{C}_{\text{motion}}$
4. Replace the corrective/instructional feedback with the automatically generated motion context, which provides:
   - Quantified form violations (*e.g.*, "Right knee flexion 85°, target 90°")
   - Anatomically precise corrective instructions (*e.g.*, "Increase knee flexion by 5° at the bottom phase")

The timestamp of each feedback instance is preserved exactly from the original QEVD-fit-coach; only the feedback content is replaced with biomechanically-grounded descriptions.

1.2. Dataset Statistics

The resulting QEVD-bio-fit-coach dataset maintains an identical structure to the original QEVD-fit-coach:

- **Training set**: 149 videos across 23 exercise types
- **Test set**: 74 videos across the same 23 exercise types

Annotation Examples. Representative transformations from the original to biomechanically-grounded feedback:

- **Squat**: "Lower your body more" → "Knee flexion 85° (target: 90°). Increase knee flexion by 5° at the bottom phase."
- **Push-up**: "Keep your back straight" → "Lumbar spine variance 12° (target: $< 5°$). Maintain neutral spine alignment; engage the core to stabilize the lumbar region."
- **Lunge**: "Don't lean forward" → "Thorax forward lean 23° from vertical. Reduce thorax flexion by 8°; keep the torso upright."
- **Plank**: "Hold steady" → "Hip angle variance 8° (target: $< 5°$). Stabilize the hip position and maintain static alignment from the shoulders to the ankles."

This automated pipeline ensures consistent, anatomically precise feedback grounded in explicit biomechanical analysis, enabling systematic evaluation of biomechanics-aware coaching systems.

2. Evaluation Metrics

We summarize the LLM-based automatic scoring pipelines; each metric uses a distinct prompt and decoding setup.

LLM-Bio-Accuracy

- **Biomech system prompt**: "You are an expert biomechanics analyst. Your role is to evaluate the factual accuracy and relevance of the biomechanical feedback generated for a user's exercise performance. Always respond as a python dictionary string."
- **Biomech user template**: `Please evaluate the following predicted biomechanical feedback:`

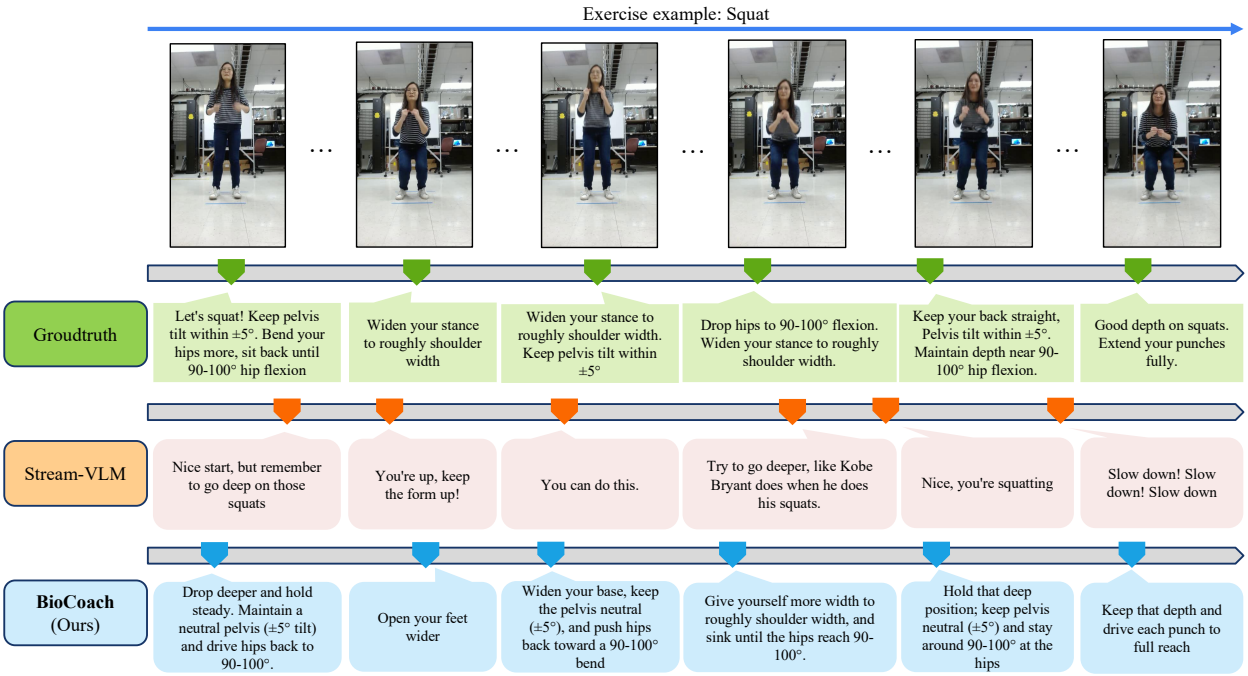

Figure 1. Qualitative timeline for a squat exercise. **BioCoach** produces temporally aligned, biomechanics-grounded cues with consistent phase tracking, while Stream-VLM outputs generic or mistimed feedback inconsistent with the ground-truth annotations.

```
-Ground truth biomechanical context:
<1>
-Predicted biomechanical feedback:
<2>
Compare the predicted biomechanical
feedback against the ground truth
context.  Focus
on maintaining factual accuracy
regarding joint angles, posture,
balance, and other factors.
biomechanical observations.  Provide
an integer accuracy score between 1
and 5
(5 = perfect alignment).  Respond
strictly as a Python dictionary
string:
{'score':  int(score)} with no extra
text.
```

3. Exercise-Specific DoF Annotation Protocol

Ground-truth salient joint sets $\mathcal{J}^{\text{gt}}$ were obtained using Qwen3-VL-30B-A3B-Instruct with biomechanics domain knowledge. Crucially, annotations are conditioned on both *exercise type* and *execution quality*. For each video segment, Qwen3-VL-30B-A3B-Instruct was provided with the exercise type (*e.g.*, squat) and the video frames. The vision-language model then identified which joints require attention by analyzing visual cues of execution quality (*e.g.*, detecting forward lean). Domain experts reviewed and validated these LLM-generated annotations. For example:

- Standard salient joints for squat = {hips, knees, ankles, lumbar}
- Expanded set for squat with forward lean = {hips, knees, ankles, lumbar, thorax}

This quality-aware annotation strategy enables the DoF selection network $\mathcal{A}_\theta$ to learn *execution-dependent* attention patterns. Unlike fixed per-exercise lookup tables, the learned selector can identify form deviations from visual cues and dynamically prioritize relevant joints, even when exercise variants or error patterns are not explicitly labeled. Similar quality-dependent annotations were obtained for all 23 exercises, with an average of $K = 10$–15 joints per video segment depending on detected form deviations.

4. Static vs. Dynamic Joint Classification

As referenced in Sec. 3.4.2 ("Biomechanical Constraint Evaluation"), each joint is classified as *static* (should remain stable) or *dynamic* (should exhibit specific motion patterns).

Classification Procedure. For each of the 23 exercise types, GPT-4 was used to classify joints as static or dynamic. The model was prompted with exercise descriptions and biomechanical principles to determine, for each joint, whether it should maintain stability or exhibit con-

trolled motion during that exercise. This produces a per-exercise lookup table of joint classifications. Domain experts reviewed and validated these LLM-generated annotations. At inference time, biomechanical constraint evaluation (Sec. 3.4.2) is applied only to the intersection of the DoF selector's predicted salient joints $\mathcal{J}^*$.

- **Static joints**: Should maintain stable angles throughout the cycle. Evaluated via variance: $\delta_j^{\text{static}} = \text{Var}(\{q_{j,i}\}_{i\in[i_s,i_e]})$. Example: spine alignment during squats should have $\delta_j^{\text{static}} < 5^\circ$
- **Dynamic joints**: Should follow specific motion patterns and reach target ranges at critical frames. Evaluated via deviation from reference at key frames: $\delta_j^{\text{dynamic}} = |q_j^{\text{user}}(i_{\text{key}}) - q_j^{\text{ref}}(i_{\text{key}})|$. Example: hip flexion during squats should reach 90°–100° at the bottom position

Exercise-Specific Examples. Below, we provide complete joint classifications for all 24 anatomical joints across representative exercises:

- **Squat**:
  - Dynamic: R/L hip, R/L knee, R/L ankle, R/L subtalar
  - Static: pelvis, lumbar, thorax, head, R/L scapula, R/L shoulder, . . . (remaining upper-body joints)
- **Push-up**:
  - Dynamic: R/L scapula, R/L shoulder, R/L elbow
  - Static: pelvis, lumbar, thorax, head, R/L hip, R/L knee, . . . (remaining lower-body joints)
- **Plank**:
  - Dynamic: none (isometric hold)
  - Static: all 24 joints (pelvis, lumbar, thorax, head, R/L hip, knee, ankle, . . .)
- **Lunge**:
  - Dynamic: R/L hip, R/L knee, R/L ankle
  - Static: pelvis, lumbar, thorax, head, R/L scapula, R/L shoulder, . . . (remaining joints)

5. Cycle Detection Algorithm Details

This section provides implementation parameters for the cycle detection algorithm described in the main paper, Sec. 3.4.2.

Video Sampling and Preprocessing. Original videos are captured at 30 fps. Gaussian smoothing is applied using `scipy.ndimage.gaussian_filter1d` with $\sigma = 2$.

Representative Joint Selection. For each exercise type, we select a representative joint whose angle trajectory exhibits the clearest periodic pattern for cycle detection. The joint is chosen empirically: squats utilize knee flexion, and push-ups involve elbow flexion. For alternating movements, we select central joints that reflect bilateral patterns (high knees use pelvis tilt), enabling cycle detection and subsequent left-right phase segmentation.

Prominence-Based Peak Detection Parameters. For repetitive exercises, we use `scipy.signal.find_peaks` with:

- Prominence threshold: $p_{\text{min}} = 0.1$ radians ($\approx 5.7^\circ$)
- Distance constraint: $d_{\text{min}} = 5$ frames (half of minimum cycle length)
- Cycle length bounds: $[24, 150]$ frames (0.8–5.0 s at 30 fps)

Alternating Movement Detection Parameters. For alternating movements (*e.g.*, high knees), after detecting cycle boundaries (i_s, i_e) via prominence-based peak detection on the central joint trajectory, zero-crossing detection is applied within the cycle to identify phase transitions. We compute the cycle-specific mean:

$$\bar{q}_{j,\text{cycle}} = \frac{1}{i_e - i_s + 1} \sum_{i=i_s}^{i_e} q_{j,i}, \tag{1}$$

and detect zero-crossings of the normalized signal $q_{j,i} - \bar{q}_{j,\text{cycle}}$ to segment left-right alternation phases. These phase boundaries enable the evaluation of bilateral limb patterns: left and right limb angles can be compared within their respective phases to assess symmetry.

Static Hold Detection Parameters. For isometric holds, rolling variance is computed with a window of $w = 10$ frames. Stability threshold $\epsilon = P_{30}(\{\sigma_t\})$ (30th percentile of rolling standard deviations). Stable segments ≥ 10 frames are extracted as hold cycles.

Reference Trajectory Source. Reference trajectories are curated using exercise-specific biomechanical rules: for each of the 23 exercises, we define canonical joint angle ranges and motion patterns based on sports science literature and domain expertise. These rules generate complete 46-dimensional angle representations with natural within-cycle variations, saved as `.npy` files. At inference, these pre-computed references serve as baselines for reference alignment and constraint evaluation.

Cycle Quality Score Computation. The four similarity metrics mentioned in main paper Sec. 3.4.2 are computed as follows: (1) all salient joint angles are concatenated and z-score normalized before computing cosine similarity; (2) Pearson correlations are averaged across salient joints $\mathcal{J}^*$; (3) frame-to-frame differences $\Delta q_{j,k} = q_{j,k+1} - q_{j,k}$ are computed for velocity comparison; (4) range of motion $\text{ROM}_j = \max(q_j) - \min(q_j)$ ratios are converted to $[0, 1]$ and averaged. The final score combines these via weighted sum: $s_{\text{cycle}} = 0.4 \cdot \text{sim}_{\text{cos}} + 0.3 \cdot \text{sim}_{\text{pearson}} + 0.2 \cdot \text{sim}_{\text{vel}} + 0.1 \cdot \text{sim}_{\text{amp}}$, clipped to $[0, 1]$. This score provides cycle quality confidence and enables best-cycle selection when multiple candidates exist.

6. Implementation Details

This section expands on the implementation overview in the main paper (Sec. 3.8), providing architecture specifications and training configurations.

6.1. DoF Selection Network Architecture

The attention network $\mathcal{A}_\theta$ (referenced in Sec. 3.3 and trained as described in Sec. 3.6) is a 3-layer MLP with learnable parameters θ that takes visual features $\mathbf{F}_t^{\text{vis}}$ as input and outputs importance scores for each joint. The network consists of:

- **Layer 1**: Projects pooled visual tokens to an intermediate 512-dimensional space with ReLU activation and a dropout rate of 0.1
- **Layer 2**: 256-dimensional hidden layer with ReLU activation and dropout rate of 0.1
- **Layer 3**: Output layer producing $J = 24$ importance scores $\mathbf{s}^t \in [0,1]^{24}$ via sigmoid activation
- **Top-K selection**: Selects top $K = 12$ joints by importance for downstream biomechanical analysis

6.2. Visual Backbone Configuration

As referenced in the main paper (Sec. 3.2, paragraph "Visual Appearance Backbone"), we employ a 3D CNN following [3].

Visual Appearance Backbone Details. Temporal window: $\tau = 12$ frames sampled at 4 Hz from the 30 fps video (3 s motion history), where each frame represents a temporal sampling point processed by the 3D CNN

Output tokens: $N_v = 35$ visual tokens with embedding dimension $d = 1280$

Architecture: 2D and 3D convolutional layers with causal masking (detailed in the main paper, Sec. 3.2). All 3D CNN weights are frozen during training; only the cross-attention layers (Sec. 7.1) are trainable

6.3. 3D Skeletal Extraction and Morphometric Processing

We extract biomechanically-grounded skeletal kinematics using HSMR [4] + SKEL [2] (referenced in Sec. 3.2, "3D Skeletal Kinematic Backbone").

46-Dimensional Skeletal Representation. Following SKEL [2], we use a 46-dimensional Euler-angle representation that defines 24 anatomical joints. The degrees of freedom (DoFs) are organized as:

- **Pelvis** (0–2): tilt, list, rotation
- **Right leg** (3–9): hip (flexion, adduction, rotation), knee angle, ankle angle, subtalar angle, mtp angle
- **Left leg** (10–16): hip (flexion, adduction, rotation), knee angle, ankle angle, subtalar angle, mtp angle
- **Spine** (17–25): lumbar (bending, extension, twist), thorax (bending, extension, twist), head (bending, extension, twist)
- **Right arm** (26–35): scapula (abduction, elevation, upward rotation), shoulder (x, y, z), elbow flexion, pro/sup, wrist (flexion, deviation)
- **Left arm** (36–45): scapula (abduction, elevation, upward rotation), shoulder (x, y, z), elbow flexion, pro/sup, wrist (flexion, deviation)

3D Skeletal Kinematic Backbone Details. Skeletal representation: 46-dimensional Euler-angle representations with joint-specific biomechanical constraints following SKEL [2]

Temporal aggregation: Shape parameters $\boldsymbol{\beta}$ averaged over the $\tau = 12$ frame window to yield stable $\bar{\boldsymbol{\beta}}$, reducing per-frame shape estimation noise

Joint angle smoothing: Gaussian smoothing is applied to joint angle trajectories to reduce jitter while preserving cycle peaks for detection

Morphometric conversion: Virtual Measurements [1] extracts `mass`, `height`, `chest`, `waist`, and `hip` circumference from the fitted SMPL mesh (procedure detailed in the main paper Sec. 3.4.1)

Confidence handling: When HSMR confidence is low (severe occlusion, extreme angles), the system retains the last valid pose and prepends warnings to $\mathcal{C}_{\text{morph}}$

6.4. Training Configuration

As noted in Sec. 3.6, we employ a fine-tuning strategy that freezes both the 3D CNN visual backbone and the LLaMA-2-7B language model. Training proceeds in two stages: first, training the DoF selection network $\mathcal{A}_\theta$, then fixing it and training the cross-attention fusion layers.

Stage 1: DoF Selection Network Pre-training. The DoF selection network $\mathcal{A}_\theta$ is first trained independently using binary cross-entropy loss $\mathcal{L}_{\text{DoF}}$ (Eq. in main paper Sec. 3.6) with ground-truth salient joint annotations $\mathcal{J}^{\text{gt}}$ obtained via Qwen3-VL-30B (detailed in Sec. 3). The network is initialized randomly and trained until the joint selection accuracy converges. Once trained, $\mathcal{A}_\theta$ is frozen to provide consistent, salient joint predictions during subsequent cross-attention training.

Stage 2: Cross-Attention Fusion Training. With $\mathcal{A}_\theta$ frozen, cross-attention projection matrices W_Q, W_K, W_V, W^O are trained via autoregressive cross-entropy loss $\mathcal{L}_{\text{CE}}$ with action-token down-weighting (Eq. in main paper Sec. 3.6). Initialized with Xavier uniform (gain $= 0.1$), the layers learn to fuse visual and morphometric features while the visual and LLM backbones remain frozen.

7. Mathematical Derivations

This section expands the mathematical formulations for vision-biomechanics conditioning (referenced in Sec. 3.5).

7.1. Vision-Morphometric Cross-Attention Details

The cross-attention mechanism fuses visual features $\mathbf{F}_t^{\text{vis}} \in \mathbb{R}^{N_v \times d}$ with morphometric context $\mathbf{m}_t \in \mathbb{R}^{N_m \times d}$.

Architecture Details.

- **Multi-head configuration**: $h = 8$ attention heads with per-head dimension $d_k = 160$
- **Initialization**: Cross-attention projection matrices W_Q, W_K, W_V, W^O initialized with Xavier uniform (gain $= 0.1$)
- **Residual connection**: $\mathbf{z}_t = \mathbf{F}_t^{\text{vis}} + \text{CrossAttn}(\cdot)$ prevents morphometric context from overriding visual evidence

7.2. Motion Context Serialization and Prompting

Motion quality context $\mathcal{C}_{\text{motion}}$ (Sec. 3.4.2 in the main paper) is serialized as structured natural language, combining pose state and detected violations.

Continuous Cycle Detection and Motion Context Updates. During streaming inference, cycle detection runs continuously: as frames arrive, the angle trajectory is updated, and the peak detection algorithm incrementally identifies completed cycles. At each timestep, the motion quality context $\mathcal{C}_{\text{motion}}$ is generated by combining (1) the current pose state $\mathbf{p}_{\text{state}}^{(t)}$, which captures instantaneous joint angles, and (2) violations $\mathbf{v}_{\text{violations}}$ from the most recently completed cycle's constraint evaluation (Sec. 3.4.2). The pose state updates every frame, while violations are updated whenever a new cycle completes. This $\mathcal{C}_{\text{motion}}$ is prepended to the prompt at each token generation step (main paper Eq. in Sec. 3.5), ensuring the LLM always conditions on up-to-date biomechanical analysis while maintaining streaming operation.

References

[1] Vasileios Choutas, Lea Müller, Chun-Hao P Huang, Siyu Tang, Dimitrios Tzionas, and Michael J Black. Accurate 3D body shape regression using metric and semantic attributes. In *CVPR*, 2022. 4

[2] Marilyn Keller, Keenon Werling, Soyong Shin, Scott Delp, Sergi Pujades, C Karen Liu, and Michael J Black. From skin to skeleton: Towards biomechanically accurate 3d digital humans. *TOG*, 2023. 4

[3] Sunny Panchal, Apratim Bhattacharyya, Guillaume Berger, Antoine Mercier, Cornelius Böhm, Florian Dietrichkeit, Reza Pourreza, Xuanlin Li, Pulkit Madan, Mingu Lee, et al. What to say and when to say it: Live fitness coaching as a testbed for situated interaction. *NeurIPS*, 2024. 1, 4

[4] Yan Xia, Xiaowei Zhou, Etienne Vouga, Qixing Huang, and Georgios Pavlakos. Reconstructing humans with a biomechanically accurate skeleton. In *CVPR*, 2025. 4