Physics-Based 3D Ball Trajectory Reconstruction from Monocular Soccer Video: A Multi-Model Benchmark

Supplementary Material

8. Appendix

Figures 3 and 4 show qualitative trajectory predictions for four models from Table 1: *basic parabola*, *basic kinetic*, *basic fitg*, and *basic angular*. Each row shows one arc segment; each column shows the same segment fitted by a different model. Segments are selected to illustrate diverse fitting outcomes: good fits, cases where one model outperforms others, and systematic failures.

8.1. Arc-loss ablation on the LP-static dataset

Table 6 ablates the three-term arc objective from Eq. (6) on the LP-static dataset. We report two representative models: *basic parabola*, the simplest arc model, and *basic fitg*, the strongest soccer model in Table 3.

Table 6. Arc-loss ablation on the LP-static dataset. We report mAP_{arc} together with the full-trajectory vertical error. Higher is better for mAP_{arc}; lower is better for error. "No end" removes $\mathcal{L}_{\text{end}}$, "no z" removes $\mathcal{L}_z$, and "traj only" keeps only $\mathcal{L}_{\text{traj}}$.

Model	Obj.	mAP_{arc}	z (m)
basic parabola	full	0.274	0.141
basic parabola	no end	0.268	0.182
basic parabola	no z	0.276	0.158
basic parabola	traj only	0.220	0.511
basic fitg	full	0.307	0.160
basic fitg	no end	0.296	0.213
basic fitg	no z	0.306	0.170
basic fitg	traj only	0.172	0.440

The same trend appears in both models. Dropping either auxiliary term already increases the vertical error, even when mAP_{arc} changes only slightly. The trajectory-only variant is clearly worse for both models. For this reason, we keep all three terms in the default objective: $\mathcal{L}_{\text{traj}}$ provides the main fit, while the endpoint and height terms make the arc predictions more reliable.

8.2. Active-play and arc/straight classification

Active-play fraction. Each dataset includes manually annotated pause intervals marking dead-ball situations, out-of-play periods, and broadcast replays. The active-play percentage (Play in Table 2) is the fraction of total sequence frames that fall outside these pause intervals. During evaluation, all paused frames are removed before metric computation.

Arc/straight classification. Trajectory segments between consecutive pivot points are classified as *arc* (airborne) or *straight* (on-ground). Each segment is first labeled by a prior fitting pass that fits both arc and straight motion models to the 3D ground truth and assigns the type with lower fitting error; these labels are stored as precomputed segment-type annotations. For segments without sufficient 3D coverage, a segment-level height fallback is applied: the mean ground-truth z-coordinate over the segment is compared against a clip-specific threshold derived by optimizing over the distribution of labeled segments. Per-frame labels are then projected from the segment-level classification onto all inter-pivot frames. The arc percentage (Arc in Table 2) reports the fraction of active-play frames belonging to arc segments.

Release. Per-frame arc/straight labels and pause annotations are publicly released for LP, SW, APIDIS, and ISSIA-3D datasets.

8.3. Ground-truth and fitting coverage

Table 7. Per-dataset ground-truth and oracle 3D fitting coverage. *GT cov.*: fraction of active-play frames with valid 3D annotations. *Oracle cov.*: fraction of active-play frames covered by accepted *basic angular* oracle-fitted segments. Δ: coverage drop in percentage points. Dataset abbreviations as in Table 2.

Dataset	GT cov. (%)	Oracle cov. (%)	Δ (pp)
LP	96.9	91.6	5.3
EB	100.0	96.3	3.7
SW	99.0	94.2	4.8
ISSIA-3D	63.2	59.9	3.3
APIDIS	95.3	46.4	48.9

Table 7 reports two coverage statistics for each dataset: the fraction of active-play frames with valid 3D ground-truth annotations (*GT coverage*) and the fraction covered by accepted *basic angular* oracle 3D fitted segments (*oracle coverage*).

For the three soccer datasets with near-complete annotations (LP, EB, SW), GT coverage exceeds 96% and the drop to oracle coverage is small, indicating that segment rejection discards only a minor fraction of available data. The consistently high oracle coverage on soccer data confirms that the pivot-point annotation methodology captures nearly all contact and bounce events during active play, and that the fitting pipeline successfully reconstructs the vast majority of the annotated trajectory. ISSIA-3D has sparser ground truth, yet the coverage drop is the smallest among

Figure 3. Qualitative model comparison on the Legia-Piast dataset (static and broadcast cameras). Each column corresponds to one physics model (Table 1). **Visualization**: red dots show the predicted 3D trajectory projected onto the video frame; green dots show the oracle 3D ground truth; blue and dark-green dots show the corresponding pitch-plane (ground) projections of the prediction and ground truth, respectively; yellow markers indicate segment endpoints (pivot points). Per-panel 3D RMSE in meters is overlaid in the bottom-right corner; the dataset and camera type are indicated in the top-left corner of the first column.

Figure 4. Qualitative model comparison on the Ekstraklasa-Broadcast (five matches) and APIDIS (basketball) datasets. Each column corresponds to one physics model (Table 1). **Visualization**: red dots show the predicted 3D trajectory projected onto the video frame; green dots show the oracle 3D ground truth; blue and dark-green dots show the corresponding pitch-plane (ground) projections of the prediction and ground truth, respectively; yellow markers indicate segment endpoints (pivot points). Per-panel 3D RMSE in meters is overlaid in the bottom-right corner; the dataset is indicated in the top-left corner of the first column.

the five datasets, suggesting that limited coverage originates in the annotation source rather than in model failure. APIDIS stands out for two reasons. First, basketball pivot annotations cover only passes and shots (Section 4.2), so active play constitutes only 7% of total frames and precise pivot placement is harder for the rapid exchanges typical of basketball. Second, the resulting segments are substantially shorter than in soccer, leaving the optimizer with fewer observations per segment to constrain the physics parameters.

8.4. MuJoCo ellipsoid fluid model

This section details the ellipsoid-based fluid force model used by the *MuJoCo ellipsoid* variant (Table 1). The total force and torque exerted by the fluid onto the ball are:

$$\begin{aligned}\mathbf{f}_{\text{ellipsoid}} &= \mathbf{f}_A + \mathbf{f}_D + \mathbf{f}_M + \mathbf{f}_K + \mathbf{f}_V \\ \mathbf{g}_{\text{ellipsoid}} &= \mathbf{g}_A + \mathbf{g}_D + \mathbf{g}_V\end{aligned} \tag{10}$$

The added-mass force and torque are:

$$\begin{aligned}\mathbf{f}_A &= -\mathbf{m}_A \circ \dot{\mathbf{v}} + (\mathbf{m}_A \circ \mathbf{v}) \times \boldsymbol{\omega} \\ \mathbf{g}_A &= -\mathbf{I}_A \circ \dot{\boldsymbol{\omega}} + (\mathbf{m}_A \circ \mathbf{v}) \times \mathbf{v} + (\mathbf{I}_A \circ \boldsymbol{\omega}) \times \boldsymbol{\omega}\end{aligned} \tag{11}$$

Quadratic drag opposes the motion and scales with the square of velocity:

$$\begin{aligned}\mathbf{f}_{\text{D}} &= -C_D \rho A_D \|\mathbf{v}\| \mathbf{v} \\ \mathbf{g}_{\text{D}} &= -C_D \rho I_D \|\boldsymbol{\omega}\| \boldsymbol{\omega}\end{aligned} \tag{12}$$

where C_D is the drag coefficient, ρ is the fluid density, and A_D is the reference surface area.

Viscous drag:

$$\begin{aligned}\mathbf{f}_{\text{V}} &= -6\pi r_D \beta \mathbf{v} \\ \mathbf{g}_{\text{V}} &= -8\pi r_D^3 \beta \boldsymbol{\omega}\end{aligned} \tag{13}$$

The Magnus force is:

$$\mathbf{f}_{\text{M}} = C_M \rho V \boldsymbol{\omega} \times \mathbf{v} \tag{14}$$

where V is the body volume and C_M is the Magnus coefficient. For a sphere, the Kutta lift $\mathbf{f}_K$ vanishes, the virtual inertia is zero, and added mass can be evaluated analytically, reducing the fitted parameters to C_D^{blunt}, C_D^{ang}, and C_M (Table 1). The projected trajectory curvature in this model arises from the Magnus force and the added-mass force, which have opposite signs.

8.5. Transformations between physics-related constants

This appendix details the mathematical transformations required to convert between different physics simulation frameworks, specifically focusing on the conversion of drag and Magnus effect parameters between a base physical model and the MuJoCo physics engine.

8.5.1. Assumptions and limitations

Throughout this analysis, we assume no virtual mass effects are present in the system. If virtual mass effects are utilized, additional corrections must be applied to account for the shift in all computed quantities due to the virtual mass modifying the applied forces and torques.

8.5.2. Drag coefficient transformations

Base model to MuJoCo conversion. The conversion between the base model's drag parameter k_3 and MuJoCo's drag coefficient C_D is analytical and well-documented in the MuJoCo documentation. The drag force in MuJoCo is expressed as:

$$\mathbf{f}_{\text{D}} = -C_D \rho A_D \|\mathbf{v}\| \mathbf{v} \tag{15}$$

where C_D is the drag coefficient, ρ is the fluid density, A_D is the reference area, and $\mathbf{v}$ is the velocity vector.

By comparing the drag force equations between the base model and MuJoCo, we obtain the following transformation relationships:

$$k_3 = C_D \cdot \rho \cdot A_D / m_{\text{ball}} \tag{16}$$

$$C_D = \frac{m_{\text{ball}}}{\rho \cdot A_D} \cdot k_3 \tag{17}$$

where m_{ball} is the mass of the ball. The division by mass in the second equation accounts for the fundamental difference in how the parameters are applied: k_3 multiplies the acceleration in the base model, while C_D multiplies the force in MuJoCo.

8.5.3. Magnus effect transformations

Model differences and challenges. The Magnus effect presents a more complex transformation challenge due to fundamental differences in how the two simulation frameworks handle this phenomenon:

- **MuJoCo implementation**: Applies a single Magnus force coefficient uniformly across all coordinate frame axes
- **Base model implementation**: Utilizes two distinct coefficients for Magnus effects applied with different magnitudes along the topspin and sidespin axes

This structural difference makes it impossible to achieve an exact parameter match between the two frameworks under general conditions.

Simplifying assumptions and approximation method. To enable approximate conversion between the frameworks, we employ the following simplifying assumptions:

1. The angular velocity remains approximately constant over time

2. The initial angular velocity in MuJoCo can be strategically chosen to control the ratio between topspin and sidespin components
3. The time-varying nature of topspin and sidespin directions (which depend on the velocity vector) is neglected under the constant angular velocity assumption

Parameter conversion procedure. The conversion procedure follows these steps:
1. Fix the Magnus force coefficient in MuJoCo
2. Determine the initial angular velocity that minimizes the three-dimensional trajectory displacement between the base model and MuJoCo simulations
3. This optimization ensures the best possible approximation given the structural differences between the two simulation frameworks

The optimization criterion can be mathematically expressed as:

$$\boldsymbol{\omega}_0^* = \arg\min_{\boldsymbol{\omega}_0} \int_0^T \|\mathbf{r}_{\text{base}}(t) - \mathbf{r}_{\text{MuJoCo}}(t, \boldsymbol{\omega}_0)\|^2 \, dt \quad (18)$$

where $\boldsymbol{\omega}_0^*$ is the optimal initial angular velocity, $\mathbf{r}_{\text{base}}(t)$ is the trajectory from the base model, $\mathbf{r}_{\text{MuJoCo}}(t, \boldsymbol{\omega}_0)$ is the MuJoCo trajectory with initial angular velocity $\boldsymbol{\omega}_0$, and T is the simulation time horizon.

8.5.4. Validation and accuracy considerations

The approximation quality depends on how well the constant angular velocity assumption holds for the specific trajectory being simulated. For trajectories with significant angular acceleration, the approximation error may become substantial, and users should validate the conversion accuracy for their specific use cases.

8.6. Modifications necessary for basketball

- All pivot timestamps are treated as high pivots in basketball.
- Leaps—neighboring trajectories with large spatial inconsistencies—are retained, as our data preparation methodology does not support their removal.
- The height penalty parameters are set as detailed in Table 8. We assume no known endpoint for basketball trajectories.

	Default		Modified
Parameter	Known endpoint	Open-ended	Value (m)
$z_{\max}$	1.2	5.0	6.0
$2z_{\max}$	2.4	10.0	12.0

Table 8. Height penalty parameters (all heights in metres): defaults from the original formulation vs. modified values.