Proceedings of the ASME 2025
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC-CIE2025
August 17-20, 2025, Anaheim, CA

DETC2025-169046

HUMAN DIGITAL TWIN FOR REALISTIC FALL SIMULATION USING THE BEHAVIORAL FOUNDATION MODEL META MOTIVO

Davide Ferrari[1,*], Caterina Rizzi[1], Andrea Vitali[1]

[1]University of Bergamo, Bergamo, Italy

ABSTRACT

Simulating human movement is essential for biomechanics, ergonomics, and machine learning applications. However, certain motions, such as falls, sudden collapses, or hazardous workplace incidents, are difficult to replicate in real conditions. Meta Motivo, a behavioral foundation model, provides a simulation framework based on reinforcement learning to simulate realistic human movements in a physics-based environment. It can be prompted to solve unseen tasks such as motion tracking, pose reaching, and reward optimization, while generating realistic simulations expressing human-like behavior. This work explores its potential in simulating challenging scenarios through a case study on falls, a critical issue in elderly care and workplace safety. Several strategies for simulating falls are presented, including custom reward functions, reaction time adjustments, and environmental modifications (e.g., slippery surfaces, obstacles, and external forces). This paper also presents two plugins developed to exploit these simulations: one for biomechanical analysis and another for the generation of a dataset for machine learning pose classification models. Although the results are promising, future refinements could further enhance the realism of falls and close-to-ground movements. However, this study highlights the broader applicability of Meta Motivo in replicating movements that are otherwise challenging to replicate experimentally.

Keywords: Digital Human Modeling, Meta Motivo, Behavioral Foundation Model, Fall Simulations, Motion Capture

1. INTRODUCTION

Human motion simulation plays a fundamental role in numerous engineering applications, including human safety, injury prevention, ergonomics, and human-robot interaction. While conventional motion capture systems (e.g., OptiTrack, Xsens, and pose detection algorithms) excel at recording standard, controlled movements, they face limitations when capturing dynamic, high-risk behaviors such as falls.

Falls have been extensively studied in various domains, particularly in elderly care [1] and workplace safety [2]. Understanding and accurately reproducing human fall dynamics is critical for assessing injury mechanisms, developing assistive technologies, and improving human-centric product design. However, traditional methods for studying falls often rely on human participants simulating falls onto protective surfaces [3, 4]. This approach has several limitations [5]:

- **Anticipation bias**: Participants alter their natural reaction and instinctively brace themselves when simulating a fall.
- **Altered impact conditions**: Falls onto soft surfaces modify acceleration patterns and contact forces, leading to discrepancies with real falls.
- **Ethical and safety constraints**: High-risk fall scenarios (e.g., stair falls, slipping on wet surfaces, or head-first impacts) cannot be ethically tested with human subjects.
- **Limited demographic representation**: Many studies primarily involve young, healthy adults, whereas populations such as older adults have distinct biomechanical responses.

Advanced computational human modeling and simulation techniques offer a viable alternative to address these issues. Recent advancements in physics-based humanoid agents have enabled the reproduction of realistic human motion in virtual environments [6]. Meta Motivo [7], released by Meta in December 2024, represents a novel approach to human motion simulation. It is a behavioral foundation model trained using an unsupervised reinforcement learning (RL) algorithm, allowing control of a physics-based humanoid across a broad range of whole-body tasks. Meta Motivo leverages observation-only motion capture datasets, enabling it to generate realistic human movement and responses to external forces.

*Corresponding author: davide.ferrari@unibg.it

In the context of fall simulation, Meta Motivo may be investigated to provide a solution by overcoming the key limitations of traditional methods. By accurately simulating natural human responses, Meta Motivo can be exploited to mitigate anticipation bias, ensure realistic impact conditions using physics-based virtual environments, and enable the study of hazardous fall scenarios that are unsafe for human participants. Furthermore, it allows the customization of digital human twins by varying biomechanical properties to simulate diverse population groups (e.g., elderly individuals and individuals with mobility impairments).

This paper aims to explore various approaches to simulate falls using Meta Motivo, including forward, backward, and lateral falls, slip-induced falls, and impacts against objects. Since Meta Motivo is not designed for biomechanical analysis and movement analysis, two custom plugins were developed to get motion capture data of the simulations. The following sections provide a review of the scientific background of fall simulation, an in-depth analysis of the different types of simulated falls, and a technical description of the developed export plugins. Finally, limitations of Meta Motivo for fall simulation are discussed, as well as future developments.

2. SCIENTIFIC BACKGROUND

Understanding human falls is a critical area of research in biomechanics, healthcare, and workplace safety. Falls are a leading cause of injury among older adults [8] and a significant risk factor in high-risk occupational settings [9]. Extensive research has been conducted on falls in both older adults and workplace environments. These studies focus on identifying risk factors, developing effective prevention strategies, understanding the broader impacts of falls, and improving fall detection technologies [10] [11]. Fall detection systems have significantly improved over the years, becoming more accurate and faster. These advancements are primarily driven by the integration of machine learning techniques and improvements in neural network architectures [12].

Various approaches to fall detection exist, including sensor-based methods that utilize accelerometers [13], often integrated into wearable sensors, and vision-based approaches that leverage state-of-the-art pose detection algorithms [14]. However, a major limitation in fall-related studies is the reliance on realistic fall data to develop accurate monitoring systems and simulations. Many fall detection algorithms are tested on datasets that may not adequately represent real-world scenarios. These datasets often contain simulated falls rather than actual recorded falls, leading to discrepancies in sensitivity and specificity, which affect real-world performance. Most datasets consist of falls performed on mattresses by young and healthy individuals, limiting the applicability of monitoring algorithms in real-world contexts [15]. Beyond fall detection, research has also focused on the biomechanical effects of falls on the human body. Numerous studies have attempted to simulate impact forces on bones through finite element analysis (FEA) [16] [17]. However, these simulations often lack specificity and, obtaining precise forces from real falls remains a significant challenge [18]. Some studies attempted to simulate falls using physics-based frameworks [19], but these approaches are constrained by the fact that falling motions are governed only by physical laws and body mechanics without realistic human reactions such as protective reflexes.

In this context, Behavior Foundation Models (BFMs) represent a recent advancement in motion simulation. These models leverage large-scale motion capture (Mocap) datasets to learn generalized movement patterns, allowing to control a human virtual agent to perform a variety of tasks without task-specific training. Reinforcement learning techniques enable these models to adapt to new conditions, making them particularly suitable for dynamic and unpredictable scenarios such as falls. Meta Motivo, introduced in December 2024 by Meta, is the first behavior foundation model based on reinforcement learning, and no prior studies have yet applied it to digital human model research, and in detail in the study of falls. Meta Motivo enables the study of fall dynamics in customizable scenarios. By adjusting environmental factors such as friction, external forces, and reaction times, these models can be leveraged to generate synthetic falls datasets for machine learning (ML) applications and impact analysis in biomechanics. The primary innovation in the context of fall simulation is that the avatar is influenced not only by physical and biomechanical constraints but also by human-like responses. Since Meta Motivo is trained on motion capture data, it can realistically emulate human reactions and adaptation strategies, such as using arms to attenuate the impact of a fall. Given the novelty of this approach, this work aims to highlight the potential of Meta Motivo in fall studies by showing different methods for simulating realistic falls using RL.

3. METHODS

Figure 1 depicts the software framework developed exploiting Meta Motivo to simulate realistic falls. Meta Motivo operates within the MuJoCo [20] (Multi-Joint dynamics with Contact) physics engine, a framework designed for high-fidelity simulation of articulated structures interacting with their environment. MuJoCo provides a physics-based environment where an articulated humanoid model responds to external forces, surfaces, objects, and environmental parameters such as gravity and friction. The combination of Meta Motivo and MuJoCo allows the simulation of human behavior within a physically realistic world. The agent is subject to the physics of its body and virtual environment, but it adapts dynamically in a human-like manner, thanks to the behavior foundation model that controls it. The Meta Motivo model is a predefined humanoid agent based on the SMPL (skinned multi-person linear model) model [21] composed of 24 rigid bodies, of which 23 are actuated.

Meta Motivo model is controlled using a reinforcement learning framework and can be prompted with a reward to optimize in order to encourage diverse and effective behaviors, allowing generalization across different tasks. In reinforcement learning, reward functions assign numerical values to different states and actions, enabling the agent to adopt behaviors that maximize its cumulative reward. This mechanism allows Meta Motivo to autonomously learn effective movement strategies through trial and error, adapting its policy over time to select actions that lead to more favorable outcomes. In this work, both predefined Meta Motivo reward functions and custom-designed rewards were explored and combined to simulate various fall scenarios. Additionally, interactions with external objects, applied forces, and

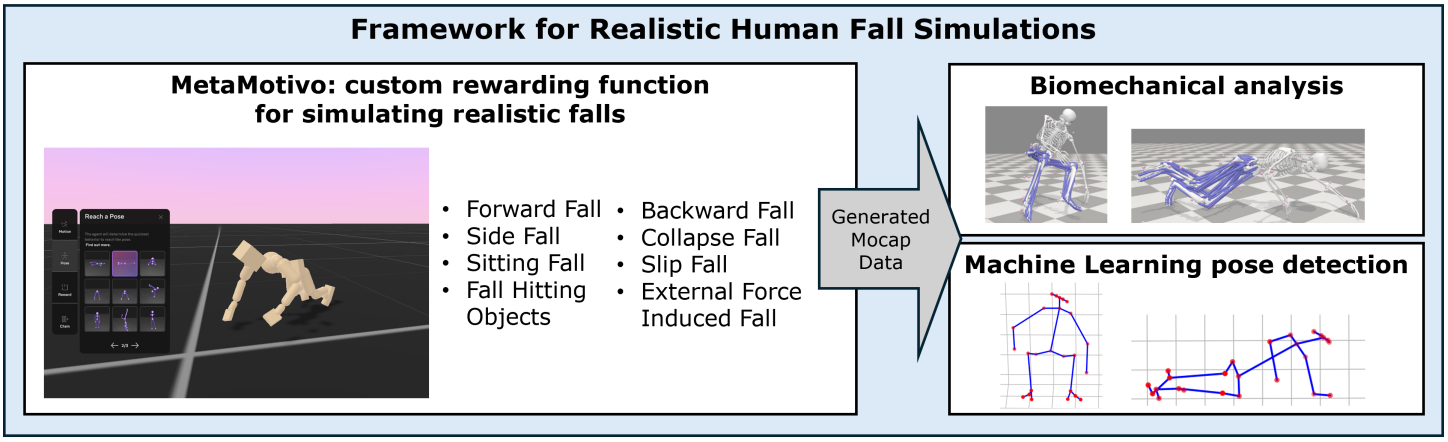


FIGURE 1: SOFTWARE ARCHITECTURE OF THE FRAMEWORK DEVELOPED FOR SIMULATING REALISTIC HUMAN FALLS STARTING FROM META MOTIVO.

varying surface frictions were introduced to enhance realism and replicate diverse real-world conditions. In detail, different type of falls were simulated:

- **Forward fall while walking**: the avatar is rewarded to fall forward while walking, simulating stumbling.
- **Side and Backward fall while standing**: the avatar is rewarded to fall backward or on its side while standing still, simulating a loss of balance.
- **Fall while sitting**:the avatar is rewarded to sit on a chair that is positioned to make it fall on the floor.
- **Fall hitting objects**: a box is positioned on the ground and the avatar is rewarder to fall forward impacting the box with the head.
- **Collapse fall**: joint actuators are reduced to simulate a muscolar weakening and a sudden collapse while standing.
- **Slip induced fall**: friction condition of the ground are changed to induce a fall while the avatar is walking, simulating a fall caused by a slipping surface.
- **External force induced fall**: a force is applied to the avatar torso simulating an external push and making it fall.

Meta Motivo does not actually provide an export option of the simulated motion. Thus, to leverage these simulated human falls in further studies, two custom plugins were developed to export motion capture data:

- **Biomechanical analysis plugin**: to allow the extraction of marker-based motion capture files. These files, similar to the output of common marker-based motion capture systems, can be used to perform biomechanical analysis with dedicated software (e.g., OpenSim, AnyBody).
- **Machine learning pose detection plugin**: to extract motion capture data as keypoints skeleton, similar to common output of pose detection machine learning algorithms (e.g., OpenPose, YOLO Pose, and MediaPipe Pose). This plugin aims at facilitating the creation of synthetic datasets for training AI-based fall detection systems using computer vision and pose estimation algorithms.

4. REWARD FUNCTIONS TO SIMULATE FALLS

In this section, different approaches to simulate falls are presented, focusing on explaining how the virtual environment was modified and which reward functions were used. In reinforcement learning-based simulations, reward functions guide the behavior of an agent, quantifying a particular state or action of the agent and driving the learning process toward actions that lead to higher rewards. A general reward function R in Meta Motivo can be expressed as:

$$R = \sum_{i=1}^{N} w_i \cdot r_i(s) \tag{1}$$

where:

- $r_i(s)$ represents an individual reward term evaluating the current agent state s.
- w_i is a weight assigned to each reward component, controlling its influence on the overall function.
- N is the total number of reward components combined in the function.

The two main strategies to modify and adapt reward function are:

1. **Weighting**: Adjusting the weights w_i allows for greater rewarding of the states with more impact on the desired final task.
2. **Thresholding**: By introducing minimum or maximum bounds, rewards can be structured to activate only when a condition is met. In Meta Motivo it is used a tolerance function defined as:

$$R = \text{tolerance}(x, (x_{\min}, x_{\max}), m, sigmoid, v_{\text{margin}}) \tag{2}$$

where:

- x is the state variable being evaluated.
- $(x_{\min}, x_{\max})$ defines the inclusive bounds for the target interval.
- m is the margin that determines how steeply the reward decreases outside the bounds (if $m = 0$, the reward is strictly zero outside the bounds).
- *sigmoid* specifies the type of decay function (e.g., 'linear', 'gaussian', 'hyperbolic').
- v_{margin} is the output value when the distance from x to the nearest bound equals m.

Through these strategies, reward functions can be adapted to simulate specific scenarios and falls. In the following subsections, each type of fall is described by including the associated reward functions, their specific reward terms, and how these were integrated with modifications to the physical environment.

4.1. Forward fall while walking

The first simulation fall approach consists of combining in sequence two different reward functions made available by Meta Motivo, `LocomotionReward` and `LieDownReward`. The `LocomotionReward` function encourages the agent to walk, and it is composed of multiple factors that work together to achieve stable locomotion:

$$R_{\text{loc}} = \text{SC} \cdot \text{SR} \cdot \text{M} \cdot \text{AR} \tag{3}$$

where:

- *Small control (SC)* penalizes excessive joint movements to encourage smooth motion.
- *Stand reward (SR)* ensures the agent remains upright while walking.
- *Move (M)* rewards forward movement.
- *Angle reward (AR)* reward movement in a desired direction.

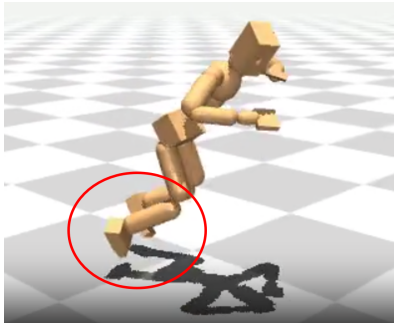
Once the walking phase is completed, the agent transitions to the `LieDownReward` function, simulating a fall. The `LieDownReward` function is used to prompt the model to lie on the floor by rewarding proximity to the ground while discouraging excessive movement. It is defined as:

$$R_{\text{lie}} = \text{SC} \cdot \text{GR} \cdot \text{DM} \cdot \text{OR} \tag{4}$$

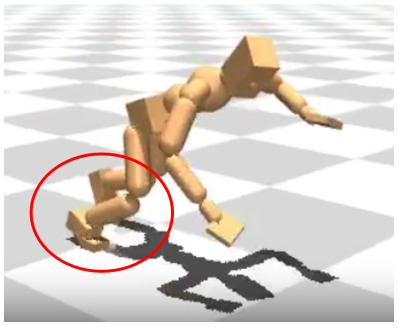
where:

- *Small control (SC)* minimizes joint torques to enable a smooth fall motion.
- *Ground reward (GR)* rewards proximity to the ground of body parts (pelvis, torso, knees).
- *Don't move (DM)* discourages excessive movement after lying down.
- *Orient reward (OR)* ensures the body maintains the correct orientation relative to the ground (i.e., prone or supine).

The simulation performs smoothly, with the hands naturally being used to mitigate impact. However, the model tends to collapse in place, pushing its legs backward rather than shifting its body forward, as shown in Figure 2a. This behavior exaggerates the fall, making it less realistic. To address this limitation, the `LieDownReward` has been modified in Python, obtaining a custom `LieDownForwardReward` that introduces a key distinction: it encourages the model to shift its center of mass ahead while falling. This is achieved by incorporating a forward movement reward for the pelvis, ensuring the model moves forward rather than simply dropping down. Additionally, foot stability constraints were introduced to promote a pivoting motion around the feet rather than knee lifting and backward pushing, resulting in a more natural falling trajectory (Fig. 2b).



(a) Original LieDownReward: the circle highlights the feet that are unnaturally lifted before falling.



(b) Modified LieDownForwardReward: The circle highlights the feet that remain on the ground making the body shifting forward.

FIGURE 2: COMPARISON BETWEEN THE ORIGINAL AND MODIFIED LIEDOWNREWARD STRATEGIES.

4.2. Side and Backward fall while standing

Like the forward fall described in the previous section, backward and side falls can also be simulated using dedicated reward functions. The (`LieDownReward`) function includes the `orient-reward` term used to determine the target orientation. When set to `"up"`, the function rewards configurations where the body parts are supine-oriented, promoting a backward fall. Conversely, setting the `orient-reward` term to `"down"`, a forward fall is encouraged, as shown in the previous section. To simulate side falls, a custom reward function `LieDownRewardSide` was developed. This function evaluates the orientation of the pelvis and its lateral rotation. By selecting the right or left side as the

direction in the reward function, the model is encouraged to position the corresponding shoulder, knee, and ankle closer to the ground, realistically simulating a side fall (Fig. 3). It is defined as:

$$R_{\text{side}} = \text{SC} \cdot \text{GR} \cdot \text{DM} \cdot \text{RR} \tag{5}$$

where:

- *Small control (SC)* minimizes joint torques to enable a smooth fall motion.
- *ground reward (GR)* rewards proximity to the ground of body parts (Shoulders, knees, ankles) of the right or left side, based on the defined side of the fall.
- *Don't move (DM)* discourages excessive movement after lying down.
- *Roll reward (RR)* ensures the pelvis rolls to the left or right based on the defined side of the fall.

By using these two functions when the avatar is standing, it is possible to simulate two simple falls, mimicking a loss of balance and a sudden hard fall on the floor.

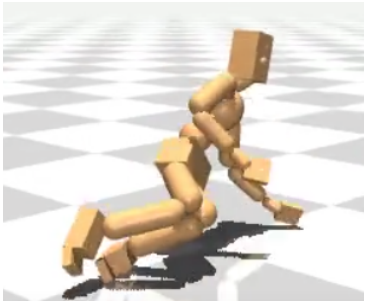


FIGURE 3: SIDE FALL.

4.3. Fall while sitting

Falls were also simulated through interactions with objects. Falls while sitting were simulated by adding a physical chair and using a reward to prompt the agent to sit. The Meta Motivo `SitOnGroundReward` was used to encourage the agent to transition to a seated position by guiding the pelvis to a lower height while maintaining an upright chest position. It ensures stability by rewarding minimal movement and proper chest orientation. It is defined as:

$$R_{\text{sit}} = \text{SC} \cdot \text{DM} \cdot \text{PR} \cdot \text{KR} \cdot \text{SR} \tag{6}$$

where:

- *Small control (SC)* minimizes joint torques to enable a smooth fall motion.
- *Don't move (DM)* discourages excessive movement after lying down.
- *Pelvis reward (PR)* ensures the pelvis remains close to the ground within a predefined threshold.
- *Knee reward (KR)* encourages the knees to maintain a constrained position or remain off the ground, depending on the defined constraint (not used in this simulation).
- *Stand reward (SR)* encourages maintaining an upright chest position during the sit-down action.

In this study, the reward was combined with a physical chair placed slightly behind and to the side of the agent, causing a loss of balance and inducing a fall (Fig. 4). After the loss of balance, the `LieDownReward` was applied to make the fall more realistic, eliminating excessive movements due to the agent trying to use the `SitOnGroundReward` even after the loss of balance.

FIGURE 4: SIT ON CHAIR FALL.

4.4. Fall hitting objects

Difficult falls to reproduce in a real-world scenario include interactions with external objects, such as hitting the body against an obstacle. This type of fall can be effectively simulated with MuJoCo and Meta Motivo by placing an object strategically in the environment so that, when the model falls, the body makes contact with it. For instance, this approach was tested using a scenario where the agent simply falls forward while walking, resulting in a collision between its head and a box placed ahead (Fig. 5). However, the results highlighted certain limitations of this method: the model protects itself only from impacts with the ground but not from the object placed ahead, which means the arms do not appropriately react to avoid or mitigate the collision. This is due to a lack of environmental awareness of the model. Beyond this example, this method can be extended to simulate other interactions with objects; however, achieving accurate falls requires precise alignment between the falling trajectory and the object position, necessitating careful planning of each trial.

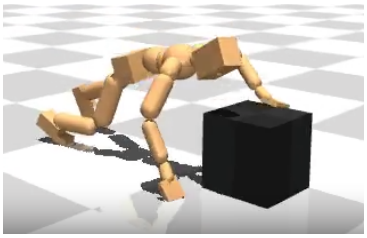


FIGURE 5: FALL HITTING THE HEAD WITH A BOX.

4.5. Collapse fall - Reduced joint control forces

All the presented fall scenarios can be further refined by adjusting the MuJoCo model reaction time and control forces. For example, reducing or temporarily removing control inputs can simulate delayed reactions, while lowering joint control forces can mimic reduced limb strength. These modifications are particularly useful for modeling demographic variability, especially in simulating falls for older people. This solution was tested by reducing both leg and arm control, inducing a collapse while standing (Fig. 6). Similarly, in a simple forward fall, the control forces were temporarily removed to delay the arm reaction and then weakened to replicate diminished arm strength, resulting in a less effective protective response to the fall.

FIGURE 6: COLLAPSE.

4.6. Slip induced fall

Falls were also simulated by varying friction conditions and trying to induce realistic falls. Specifically, floor friction was reduced mid-simulation to cause the model's feet to slip, leading to a loss of balance. The `LocomotionReward` was initially used to simulate walking, and after a few frames, the floor friction was lowered to trigger the fall. To prevent excessive compensatory movements from the model attempting to maintain locomotion, the `LieDownReward` in the supine direction was used to guide the simulation toward a controlled conclusion (Fig. 7). To enhance realism, friction was restored shortly after the loss of balance, preventing unnatural behaviors of the model.

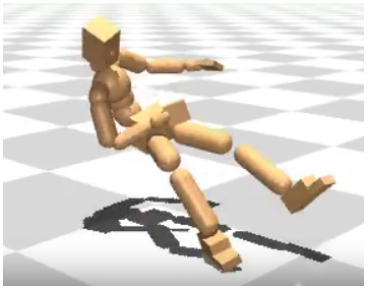


FIGURE 7: SLIP INDUCED FALL.

4.7. External force induced fall

External force-induced falls were recreated to simulate a push on the model. The primary challenge was the rapid and efficient reaction of the model to external forces, which quickly counterbalanced the pushes in a way that appeared unnatural. Consequently, applying a simple force to the torso failed to induce a fall, unless forces exceeding 500N were used. However, such large forces are unrealistic for everyday scenarios, where a fall could result from an accidental push. To address this limitation, leg control parameters were reduced immediately after applying the push to minimize the agent's ability to counterbalance and restore stability, effectively inducing a fall. For example, Figure 8 shows a simulation frame in which the agent was initially walking when a force was applied laterally to the torso, combined with a 50% reduction in the leg controls. Although visually similar to previously demonstrated falls, this external force interaction generates a distinct fall pattern characterized by the sudden acceleration of the torso due to the push.

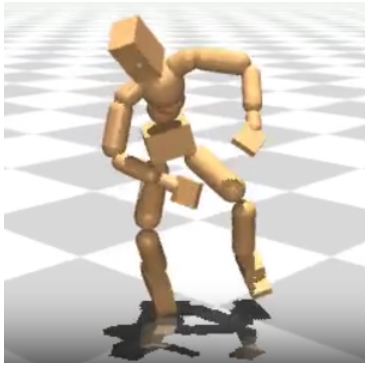


FIGURE 8: FALL INDUCED BY A PUSH.

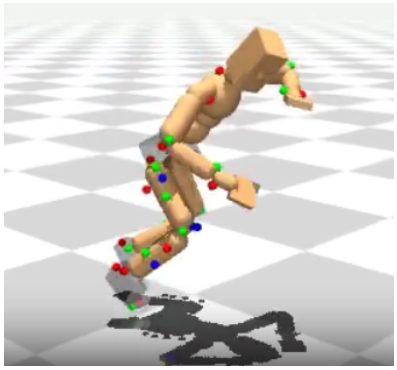
5. PLUGINS FOR MOTION ANALYSIS

To use the previously proposed simulations in motion capture studies of falls, two different plugins were developed to export the kinematics. The first one allows the generation of a marker-based file that can be used to perform inverse Kinematics with biomechanical analysis software. The latter allows the export of motion capture as a keypoint skeleton, typical of computer vision pose detection algorithms.

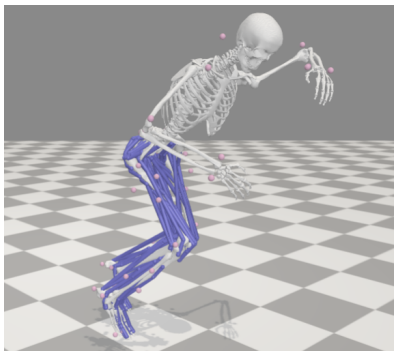
5.1. Biomechanical analysis plugin

This plugin was developed to use fall motion capture inside biomechanical simulation software. While kinematic data alone may offer limited insights into fall dynamics, biomechanical simulation software provides advanced tools for analyzing fall dynamics, including impact force simulations with objects and the ground. To generate a marker-based file of the fall, 43 markers were applied to the avatar using the MuJoCo `site` objects. These `site` objects serve as spatial reference points that can be attached to the model without influencing its kinematics (Fig. 9a). During the simulation, the 3D positions of these `sites` are recorded frame by frame and stored in an output file. Marker positions serve as essential reference points, allowing the inverse kinematics algorithm to reconstruct the avatar's movements by aligning musculoskeletal model markers with the Meta Motivo avatar markers. The developed workflow was validated using OpenSim [22], an open-source platform specializing in musculoskeletal modeling and dynamic simulations. Initially,

the OpenSim biomechanical model was scaled to match the dimensions of the Meta Motivo avatar. Subsequently, the scaled model underwent inverse kinematics analysis utilizing the generated marker-based file (Fig. 9b).



(a) Meta Motivo model with sites used as markers.



(b) OpenSim Inverse Kinematics of the simulated fall.

FIGURE 9: USING OPENSIM WITH META MOTIVO

5.2. Machine learning pose detection plugin

The keypoints plugin was developed to export the kinematics of the Meta Motivo model as a keypoint skeleton, analogous to those typically generated by machine learning-based pose detection algorithms (e.g., YOLO pose, MediaPipe Pose, OpenPose). Specifically, the BODY_25 model from OpenPose [23] was selected as a reference for this study, consisting of 25 keypoints representing major anatomical landmarks: 5 for the face, 7 for the upper body, and 13 for the lower body and feet. Similar to the marker-based plugin, `site` objects were added to the Meta Motivo model at positions corresponding precisely to the BODY_25 model keypoints. Three-dimensional coordinates of these keypoints are recorded frame-by-frame and exported as a CSV file (Fig.10). Unlike traditional pose detection algorithms, which typically derive keypoint positions directly from RGB video streams in two dimensions, this plugin generates accurate ground-truth keypoint positions in three-dimensional space directly from the simulation. However, these 3D points can be projected onto different camera views, effectively simulating multiple 2D perspectives of the same fall or motion event. This capability enables the creation of versatile datasets that can be employed in training and validating pose detection and monitoring algorithms relying on 2D skeleton extraction from RGB camera images.

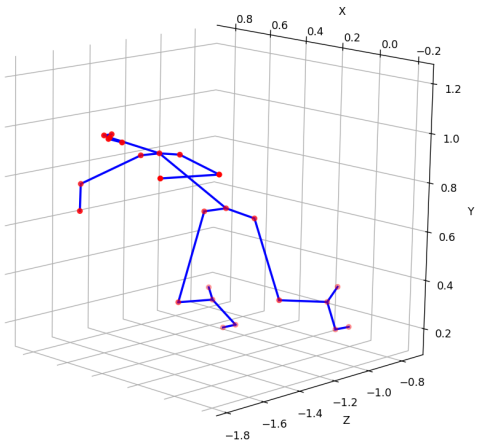


FIGURE 10: OPENPOSE SKELETON WHILE FALLING.

6. CONCLUSION

This study explores the potential of Meta Motivo in simulating fall scenarios, addressing the limitations of traditional approaches. By using reinforcement learning and physics-based modeling, various types of falls were simulated, demonstrating how different scenarios can be reproduced through reward functions, environmental modifications, and model force actuators modifications. The results obtained highlight how Meta Motivo can be a valuable tool for the generation of synthetic fall data. Such data can be used for biomechanical fall analysis and the creation of extensive datasets to train machine learning models for fall detection and prediction. To enhance the usability of the simulated falls, two plugins for extracting motion data were developed. The first retrieves marker-based motion data, making it compatible with biomechanical analysis software, while the latter extracts a keypoint, skeleton compatible with pose estimation models. Despite the promising results, limitations emerged. One major constraint is the lack of environmental awareness in the current model, preventing it from recognizing or reacting to obstacles, as observed in the "fall hitting object" scenario. Another limitation is the difficulty in simulating tasks that involve significant height changes. Attempts to model falls from elevated platforms resulted in unnatural avatar responses. Future improvements should focus on simulating a more diverse population by varying body weight, height, and strength and introducing differences in upper and lower limb reactivity to better represent elderly individuals with delayed reflexes. Moreover, a validation approach should be formulated to validate the realism of the simulated falls. However, the framework presents significant potential for studying falls and overcoming existing limitations. The ability to generate large-scale fall datasets could contribute to the development of more robust fall detection systems with better generalization across different fall types. Furthermore, these simulations could be used to study injury prevention and aid in the design of protective systems.

ACKNOWLEDGMENTS

This research work was partially funded by Regione Lombardia (Italy), regional law n° 9/2020, resolution n° 3776/2020 and by the National Plan for NRRP Complementary Investments

(PNC, established with the decree-law 6 May 2021, n. 59, converted by law n. 101 of 2021) in the call for the funding of research initiatives for technologies and innovative trajectories in the health and care sectors (Directorial Decree n. 931 of 06-06-2022) - project n. PNC000000 - AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM). This work reflects only the authors' views and opinions, neither the Ministry for University and Research nor the European Commission can be considered responsible for them.

REFERENCES

[1] Ambrose, Anne Felicia, Paul, Geet and Hausdorff, Jeffrey M. "Risk factors for falls among older adults: A review of the literature." *Maturitas* Vol. 75 No. 1 (2013): p. 51 – 61. DOI 10.1016/j.maturitas.2013.02.009. Cited by: 1282.

[2] Pereira, Filipa, González García, María de las Nieves and Poças Martins, João. "BIM for Safety: Applying Real-Time Monitoring Technologies to Prevent Falls from Height in Construction." *Applied Sciences* Vol. 15 No. 4 (2025). DOI 10.3390/app15042218.

[3] de Lourdes Martínez-Villaseñor, María, Ponce, Hiram, Brieva, J., Moya-Albor, E., Núñez-Martínez, José and Peñafort-Asturiano, Carlos. "UP-Fall Detection Dataset: A Multimodal Approach." *Sensors (Basel, Switzerland)* Vol. 19 (2019). DOI 10.3390/s19091988.

[4] Sucerquia, Angela, Lopez, Jose and Vargas-Bonilla, J. "SisFall: A Fall and Movement Dataset." *Sensors* Vol. 17 (2017): p. 198. DOI 10.3390/s17010198.

[5] Saleh, Majd, Abbas, Manuel and Le Jeannès, Régine Bouquin. "FallAllD: An Open Dataset of Human Falls and Activities of Daily Living for Classical and Deep Learning Applications." *IEEE Sensors Journal* Vol. 21 No. 2 (2021): pp. 1849–1858. DOI 10.1109/JSEN.2020.3018335.

[6] Chiu, Yi-Hung, Lee, Ung Hee, Song, Changseob, Hu, Manaen and Kang, Inseung. "Learning Speed-Adaptive Walking Agent Using Imitation Learning with Physics-Informed Simulation." (2024). DOI https://doi.org/10.48550/arXiv.2412.03949. URL https://arxiv.org/abs/2412.03949.

[7] Tirinzoni, Andrea, Touati, Ahmed, Farebrother, Jesse, Guzek, Mateusz, Kanervisto, Anssi, Xu, Yingchen, Lazaric, Alessandro and Pirotta, Matteo. "Zero-Shot Whole-Body Humanoid Control via Behavioral Foundation Models." (2025). DOI https://doi.org/10.48550/arXiv.2504.11054. URL https://arxiv.org/abs/2504.11054.

[8] Moreland, B., Kakara, Ramakrishna S. and Henry, Ankita. "Trends in Nonfatal Falls and Fall-Related Injuries Among Adults Aged 65 Years — United States, 2012–2018." *Morbidity and Mortality Weekly Report* Vol. 69 (2020): pp. 875 – 881. DOI 10.15585/mmwr.mm6927a5.

[9] Wah, Win, Berecki-Gisolf, Janneke and Walker-Bone, Karen. "Epidemiology of work-related fall injuries resulting in hospitalisation: individual and work risk factors and severity." *Occupational and Environmental Medicine* Vol. 81 No. 2 (2024): pp. 66–73. DOI 10.1136/oemed-2023-109079. URL https://oem.bmj.com/content/81/2/66.full.pdf.

[10] Xu, Qin, mei Ou, Xue and Li, Jinfeng. "The risk of falls among the aging population: A systematic review and meta-analysis." *Frontiers in Public Health* Vol. 10 (2022). DOI 10.3389/fpubh.2022.902599.

[11] Ren, Lingmei and Peng, Yanjun. "Research of Fall Detection and Fall Prevention Technologies: A Systematic Review." *IEEE Access* Vol. 7 (2019): pp. 77702–77722. DOI 10.1109/ACCESS.2019.2922708.

[12] Newaz, Nishat Tasnim and Hanada, Eisuke. "The Methods of Fall Detection: A Literature Review." *Sensors* Vol. 23 No. 11 (2023). DOI 10.3390/s23115212. Cited by: 33; All Open Access, Gold Open Access, Green Open Access.

[13] Mauldin, Taylor R., Canby, Marc E., Metsis, Vangelis, Ngu, Anne H. H. and Rivera, Coralys Cubero. "SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning." *Sensors* Vol. 18 No. 10 (2018). DOI 10.3390/s18103363.

[14] Salimi, Mohammad, Machado, J. and Tavares, J. M. R. "Using Deep Neural Networks for Human Fall Detection Based on Pose Estimation." *Sensors (Basel, Switzerland)* Vol. 22 (2022). DOI 10.3390/s22124544.

[15] Usmani, Sara, Saboor, Abdul, Haris, Muhammad, Khan, Muneeb A. and Park, Heemin. "Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review." *Sensors* Vol. 21 No. 15 (2021). DOI 10.3390/s21155134.

[16] HIRABAYASHI, Satoko, TSUCHIDA, Tomonari, TANAKA, Eiichi and MIZUNO, Koji. "Simulation Study on Mechanisms of Hip Fractures in Backward Falls." *Journal of Biomechanical Science and Engineering* Vol. 8 No. 4 (2013): pp. 328–343. DOI 10.1299/jbse.8.328.

[17] Park, Jinhee and Nam, Yun Ja. "Finite element modeling and simulation of hip joints in elderly women: for development of protective clothing against fracture." *International Journal of Clothing Science and Technology* Vol. 32 No. 5 (2020): p. 661 – 675. DOI 10.1108/IJCST-09-2019-0140. Cited by: 4.

[18] Sarvi, M. N. and Luo, Yunhua. "Sideways fall-induced impact force and its effect on hip fracture risk: a review." *Osteoporosis International* Vol. 28 (2017): pp. 2759–2780. DOI 10.1007/s00198-017-4138-5.

[19] Mastorakis, Georgios, Ellis, Tim and Makris, Dimitrios. "Fall detection without people: A simulation approach tackling video data scarcity." *Expert Systems with Applications* Vol. 112 (2018): pp. 125–137. DOI https://doi.org/10.1016/j.eswa.2018.06.019.

[20] Todorov, Emanuel, Erez, Tom and Tassa, Yuval. "MuJoCo: A physics engine for model-based control." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*: pp. 5026–5033. 2012. DOI 10.1109/IROS.2012.6386109.

[21] Luo, Zhengyi. "SMPLSim: Simulating SMPL/SMPLX Humanoids in MuJoCo and Isaac Gym." https://github.com/ZhengyiLuo/SMPLSim (2023).

[22] Delp, Scott L., Anderson, Frank C., Arnold, Allison S., Loan, Peter, Habib, Ayman, John, Chand T., Guendel-

man, Eran and Thelen, Darryl G. "OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement." *IEEE Transactions on Biomedical Engineering* Vol. 54 No. 11 (2007): pp. 1940–1950. DOI 10.1109/TBME.2007.901024.

[23] Cao, Zhe, Hidalgo, Gines, Simon, Tomas, Wei, Shih-En and Sheikh, Yaser. "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields." *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 43 No. 1 (2021): p. 172–186. DOI 10.1109/TPAMI.2019.2929257.