Neural Monocular 3D Human Motion Capture with Physical Awareness

SOSHI SHIMADA, Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
VLADISLAV GOLYANIK, Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
WEIPENG XU, Facebook Reality Labs, USA
PATRICK PÉREZ, Valeo.ai, France
CHRISTIAN THEOBALT, Max Planck Institute for Informatics, Saarland Informatics Campus, Germany

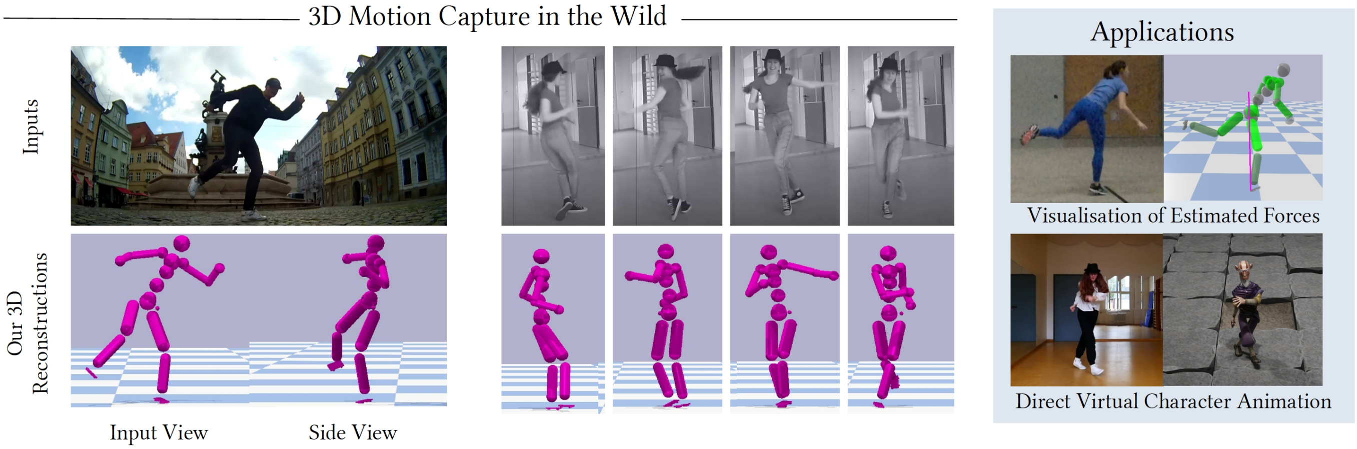
3D Motion Capture in the Wild
Inputs
Our 3D Reconstructions
Input View
Side View
Applications
Visualisation of Estimated Forces
Direct Virtual Character Animation

Fig. 1. From an input monocular video, our method for markerless 3D human motion capture estimates global human poses which obey (bio-)physical constraints. In contrast to existing methods with physical awareness, our approach is neural and fully differentiable; it allows learning motion priors and the associated physical properties from the data. We can reconstruct more challenging and faster motions compared to the state of the art, with fewer artefacts such as jitter, foot-floor penetration and unnatural body postures. Thanks to these properties, our method can be used to directly drive a virtual character or visualise joint torques. (Left:) Results of our method on different sequences from the input and side views. (Right:) Applications in motion analysis by force visualisation and virtual character animation.

We present a new trainable system for physically plausible markerless 3D human motion capture, which achieves state-of-the-art results in a broad range of challenging scenarios. Unlike most neural methods for human motion capture, our approach, which we dub "physionical", is aware of physical and environmental constraints. It combines in a fully-differentiable way several key innovations, *i.e.*, 1) a proportional-derivative controller, with gains predicted by a neural network, that reduces delays even in the presence of fast motions, 2) an explicit rigid body dynamics model and 3) a novel optimisation layer that prevents physically implausible foot-floor penetration as a hard constraint. The inputs to our system are 2D joint keypoints, which are canonicalised in a novel way so as to reduce the dependency on intrinsic camera parameters—both at train and test time. This enables more accurate global translation estimation without generalisability loss. Our model can be finetuned only with 2D annotations when the 3D annotations are not available. It produces smooth and physically-principled 3D motions in an interactive frame rate in a wide variety of challenging scenes, including newly recorded ones. Its advantages are especially noticeable on in-the-wild sequences that significantly differ from common 3D pose estimation benchmarks such as Human 3.6M and MPI-INF-3DHP. Qualitative results are provided in the supplementary video.

CCS Concepts: • **Computer methodologies → Computer graphics**; • **Motion capture**;

Additional Key Words and Phrases: Monocular 3D Human Motion Capture, Physical Awareness, Global 3D, Physionical Approach.

ACM Reference Format:
Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt. 2021. Neural Monocular 3D Human Motion Capture with Physical Awareness. *ACM Trans. Graph.* 40, 4, Article 83 (August 2021), 15 pages. https://doi.org/10.1145/3450626.3459825

1 INTRODUCTION

3D human motion capture is an actively researched area enabling many applications ranging from human activity recognition to sports analysis, virtual-character animation, film production, human-computer interaction and mixed reality. Since marker-based and

Authors' addresses: Soshi Shimada, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, sshimada@mpi-inf.mpg.de; Vladislav Golyanik, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, golyanik@mpi-inf.mpg.de; Weipeng Xu, Facebook Reality Labs, Pittsburgh, USA, xuweipeng@fb.com; Patrick Pérez, Valeo.ai, Paris, France, patrick.perez@valeo.com; Christian Theobalt, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, theobalt@mpi-inf.mpg.de.

0730-0301/2021/8-ART83
https://doi.org/10.1145/3450626.3459825

multi-camera-based solutions are expensive and unsuited for many applications (*e.g.*, in-the-wild capture and recordings outside the studio or legacy content), methods for *markerless* 3D human motion capture from a monocular camera [Mehta et al. 2017b; Shimada et al. 2020] are intensively researched.

Monocular 3D human motion capture is a highly challenging inverse problem, due to the fundamental ambiguities in deducing 3D body configuration and scale from 2D cues, as well as due to difficult (self-)occlusions [Kocabas et al. 2020a; Martinez et al. 2017; Mehta et al. 2017b; Pavlakos et al. 2018]. Most state-of-the-art methods for 3D human pose estimation and motion capture benefit from the rapid progress in machine learning and have shown stark improvements in accuracy [Kocabas et al. 2020a; Sun et al. 2019; Wandt and Rosenhahn 2019]. Despite this progress, existing predominantly purely kinematic methods still have important limitations and produce notable artefacts. Many produce per-frame predictions that can be temporally highly unstable, and many produce root-relative but not global 3D poses. Further, most existing methods are incapable by design to consider interactions with the environment, let alone biophysical pose or motion plausibility. The former often leads to collision violations such as foot-floor penetration and floating in the air in captured motions, the latter yields impossible poses with physically-impossible leaning and posture, or poses that would actually cause loss of balance. Captured results are therefore not only inaccurate in several ways but also unnatural, which greatly reduces data usability, in particular in computer graphics related applications.

We, therefore, propose a new neural network-based approach for monocular 3D human motion capture which considers physical constraints in the observed scenes, see Fig. 1 for an overview.

We believe that improving upon the recently proposed ideas of physical-awareness constraints in monocular 3D human motion capture [Rempe et al. 2020; Shimada et al. 2020] and combining them with machine learning techniques can lead to further advances in the domain. While the methods of [Rempe et al. 2020; Shimada et al. 2020] contain two stages—with the physics-based pose optimisation implemented as an engineered method relying on classical optimisation techniques,—*we are the first to propose a fully-differentiable framework for monocular 3D human motion capture with physical awareness*. Thus, our physics-based pose optimisation is a trainable neural network with custom layers for physics-based constraints. We refer to our approach as *physionical*, which means that it is fully-differentiable, neural network-based and aware of physical boundary conditions. The 3D motions estimated by our framework are smooth and natural, and can directly drive an animation character with no further postprocessing. We can also visualise the joint torques and ground reaction forces estimated from the motion in the video, which can be used for some applications, *e.g.*, sports analysis. See Fig. 1 for the visualisation of the reconstructed 3D motions and the example applications of our framework.

Our method includes two core neural components, *i.e.*, a *target pose estimator network* (TPNet) and an iterative *dynamic cycle* for controlling a humanoid character while considering physics-based boundary conditions. Both TPNet and the dynamic cycle are newly developed neural networks that are end-to-end trained. TPNet kinematically regresses the target reference 3D pose from input 2D keypoints that are obtained by an off-the-shelf 2D detector, which serves as a foundation for the dynamic cycle. The dynamic cycle first calculates gain parameters of a neural proportional-derivative (PD) controller which generates a force vector to control the kinematic character with physics properties through the differentiable physics model. The force vector is then used to estimate the ground reaction force (GRF), and both are then passed to the forward dynamics module which regresses the accelerations of the skeleton. The latter are subsequently used to update the final global human pose in 3D which matches the subject's 2D pose in the input frames and obeys the condition of plausible foot-floor placements. In the dynamic cycle our architecture contains a novel differentiable layer realising a *hard* constraint for preventing foot-floor penetration. Our motivation for a custom optimisation layer comes from the fact that conventional losses in neural networks can only express soft constraints on the learned manifold, *i.e.*, there is no guarantee that the expressed boundary conditions will be strictly fulfilled at inference time. On the other hand, physical constraints and forces such as gravity and ground reaction force (originating from the floor which naturally limits human motions) are strictly present in the physical world without freedom of interpretation.

Since our architecture is fully differentiable, it is the first approach for monocular physics-aware 3D motion capture that can be equally trained on images annotated with strong and weak labels, *i.e.*, joint angles, 3D joint keypoints but also 2D joint keypoints. Since also 2D training data can be used, our method can be trained for better generalisation and is easier to fine-tune for motion classes for which any 3D annotation would be very hard (*e.g.*, in-the-wild athletics or sports videos). Our physionical method is aware of the environment and physical laws and runs in real time at 20 frames per second. It outputs physically-plausible results with significantly fewer artefacts—such as unnatural temporal instabilities and frame-to-frame jitter, foot-floor penetration and the uncertainty in the absolute human poses along the depth dimension—than purely kinematic methods and other physics-aware methods. Moreover, compared to the previous most related method PhysCap [Shimada et al. 2020], we mitigate the delay between the observed and estimated motions. To summarise, the technical **contributions** of this article are as follows:

- The first entirely-neural and fully-differentiable approach for markerless 3D human motion capture from monocular videos with physics constraints, which we call physionical (Sec. 3).
- A new canonicalisation of input 2D keypoints allowing network training and 3D human pose regression with different intrinsic camera parameters and jointly on several datasets (Sec. 3.2). In contrast to existing normalisation methods for human pose estimation in the literature, our canonicalisation does not discard the cues for the global pose estimation.
- The integration of hard boundary conditions in our architecture to prevent foot-floor penetrations by taking advantage of the recent progress in designing optimisation layers for neural architectures [Agrawal et al. 2019b] (Sec. 3.4).
- Applications of our method in direct virtual character animation and visualisation of joint torques related to muscle activation

forces, which can be used to analyse the captured motions in conceivable downstream tasks (Sec. 3.7).

The proposed method establishes a new state of the art and outperforms existing methods on several metrics, as shown in our experiments (Sec. 4). We evaluate it on several datasets including Human3.6M [Ionescu et al. 2013], MPI-INF-3DHP [Mehta et al. 2017a], DeepCap [Habermann et al. 2020] as well as newly-recorded sequences (Sec. 4). The differences in the results of our physionical approach compared to existing techniques are especially noticeable when they are obtained on scenes in the wild. See our supplementary video with visualisations of the experimental results.

2 RELATED WORK

A vast body of literature is devoted to 3D human motion capture with multi-view systems [Bo and Sminchisescu 2008; Brox et al. 2010; Elhayek et al. 2015; Gall et al. 2010; Martin-Brualla et al. 2018; Starck and Hilton 2007; Wu et al. 2012] and inertial on-body sensors [Dejnabadi et al. 2006; Tautges et al. 2011; Vlasic et al. 2007; von Marcard et al. 2017]. Both areas are well studied and these methods have shown impressive results. On the downside, they require specialised camera rigs and hardware which make their operation outside the studio difficult. In this section, we thus further focus on related works on 1) physics-based virtual character animation and 2) monocular 3D human pose estimation and motion capture.

Physics-Based Virtual Character Animation. Many works have been proposed for physics-based character animation which is a significantly different problem compared to monocular 3D human motion capture. In virtual character animation, there is full control over the simulated physical laws and the structure of the simulated world (in which virtual characters are moving), whereas we are interested in reconstructing physically-plausible human motions from partial observations (monocular videos). At the same time, the animated character of these methods is inspirational for us, as they provide the realism and motion plausibility of character motion required in computer graphics applications [Andrews et al. 2016; Barzel et al. 1996; Bergamin et al. 2019; Levine and Popović 2012; Liu et al. 2010; Sharon and van de Panne 2005; Wrotek et al. 2006; Zheng and Yamane 2013]. Some techniques for virtual character animation employ deep reinforcement learning and motion imitation in physics engines, often requiring specialised networks for each motion kind [Bergamin et al. 2019; Jiang et al. 2019; Lee et al. 2019; Peng et al. 2018a,b]. In contrast to the latter, our problem requires a different approach. Since our goals are the generalisability across different motions and high data throughout enabling real-time applications, we use explicit equations of motions and physics-based constraints on top of initial kinematic estimates, while preserving the differentiability of our architecture trained in a supervised manner.

Classical Monocular 3D Human Motion Capture and Pose Estimation. This section focuses on the majority of works on monocular 3D human motion capture and pose estimation that do not use explicit physics-based and environment constraints. All such methods for 3D human pose estimation and motion capture can be classified into 1) direct regression approaches, 2) lifting approaches and 3) various hybrid approaches leveraging mixtures of 3D and 2D predictions. The first category of methods is based on convolutional neural networks and regresses 3D joints directly from input images [Mehta et al. 2017a; Rhodin et al. 2018; Tekin et al. 2016]. The methods of the second category regress 3D joints from detected 2D keypoints [Chen and Ramanan 2017; Martinez et al. 2017; Moreno-Noguer 2017; Pavlakos et al. 2018; Tomè et al. 2017]. Finally, multiple methods combine 3D joint depth (or location probabilities) and 2D keypoint prediction with lifting constraints [Habibie et al. 2019; Mehta et al. 2017b; Newell et al. 2016; Pavlakos et al. 2017; Yang et al. 2018; Zhou et al. 2017]. Among them, VNect [Habibie et al. 2019; Mehta et al. 2017b] uses additional weak supervision with in-the-wild images.

Some methods additionally use 3D shape priors. Statistical human body models provide strong constraints on plausible human postures which can be used for human pose estimation [Bogo et al. 2016; Kanazawa et al. 2018; Kocabas et al. 2020a]. [Habermann et al. 2020; Xu et al. 2020; Xu et al. 2018] leverage actor-specific 3D human body templates for global 3D human motion capture with shape tracking including surface deformations on top of a skeletal motion. Several further algorithms use different variants of anatomical constraints for the human body (*e.g.*, body symmetry) and show improved results in weakly-supervised [Dabral et al. 2018; Wandt and Rosenhahn 2019] or even unsupervised 3D human pose estimation [Kovalenko et al. 2019]. [Hassan et al. 2019; Zhang et al. 2020] use geometric vicinity and collision avoidance constraints for the reconstruction of human-object interactions, and [Dabral et al. 2019; Fabbri et al. 2020; Mehta et al. 2020; Rogez et al. 2019; Zanfir et al. 2018] can generalise to multiple persons in the scene.

Most of the proposed algorithms work on single images [Kanazawa et al. 2018; Kolotouros et al. 2019; Pavlakos et al. 2018; Song et al. 2020; Sun et al. 2019], whereas others take the temporal information into account for improved temporal stability [Kanazawa et al. 2019; Kocabas et al. 2020b; Pavllo et al. 2019]. To directly drive a kinematic character with skinned rigs, we need joint angles, root translation and rotation of a consistent skeleton. Only few works estimate those from the input RGB video directly and realise the character motion control from a video [Mehta et al. 2020, 2017b; Shi et al. 2020]. Among the latter, MotioNet of Shi *et al.* [2020] is the most closely related method to ours. Unlike our approach, it does not include an explicit physics model, which adds up to physically-implausible effects in the estimates. Upon the architecture design, MotioNet expects at testing the same intrinsic camera parameters as in the training dataset, *i.e.*, when the system is applied to sequences with different camera intrinsics, the accuracy of the estimated translations can vary considerably. In contrast, we use canonical 2D keypoints which makes our physionical approach invariant to camera intrinsics.

Monocular 3D Human Motion Capture with Physics-based Constraints. This section focuses on the emerging field of monocular 3D human motion capture with physics-based constraints. One of the pioneering works in this domain was proposed by Wei and Chai [2010] back in 2010. Their method requires manual user interactions for each input sequence and is computationally expensive. Vondrak et al. [2012] perform 3D human motion capture from monocular videos for physically-plausible character control. They recover 3D bipedal controllers using optimal control theory, which are capable of simulating the observed motions in different environments. Unfortunately, this method cannot easily generalise across motions and does not run in real time. Zell et al. [2017] estimate 3D human poses along with the inner and exterior forces from images for object lifting and walking. Li et al. [2019] regress human and object poses in 3D along with forces and torques exerted by human limbs from a monocular video and an object prior. They focus on instruments with grips and recognise contacts between a person and an object (*i.e.*, the instrument or the ground) to facilitate the trajectory-optimisation problem. The method of Zell et al. [2020] for the analysis of 3D human motion capture relates to our setting. It infers ground-reaction forces and joint torques from input 3D human motion capture sequences. It relies on a new dataset with multiple human motion types and ground-truth forces acquired using force plates on the floor. The advantage of this method is that the proposed forward and inverse dynamics layers generalise to new locomotion types. Thus, the main focus lies on the *explainability* of the captured human motions in 3D from the physics perspective, whereas our goal is 3D human motion capture that satisfies physics-based (environmental) constraints at interactive framerates.

Two recent methods for monocular 3D human motion capture with physics constraints are [Rempe et al. 2020] and [Shimada et al. 2020]. They tackle general human motions by introducing laws of physics as regularisers in their formulations. Both methods 1) start with initial kinematic estimates ([Xiang et al. 2019] and [Mehta et al. 2017b], respectively) which are subsequently refined through physics-based optimisation, 2) detect foot contacts and 3) assume that orientation of the ground plane is known (the final position can be refined), the camera is not moving and the entire human body is visible in all frames. [Rempe et al. 2020] and [Shimada et al. 2020], however, differ significantly in physics-based global pose optimisation and the overall runtime. Rempe et al. [2020] use as a proxy a reduced-dimensional model of the lower body inspired by [Winkler et al. 2018], which does not include all joints but captures the overall motion and contacts. In contrast, Shimada et al. [2020] rely on initial kinematic pose corrections and a lightweight iterative physics-based pose refinement with PD joint controllers and ground-reaction force estimation, which enable real-time operation. Both these approaches are compositional and only partially rely on neural networks (for the kinematic estimates and foot contact detections, but not for the physics-based reasoning), unlike our approach. We embed hard physics-based constraints through a custom layer in our architecture [Agrawal et al. 2019a] and enable its full differentiability. Our trainable model with explicit physics-based constraints realises more plausible 3D motion qualitatively and more accurate 3D poses quantitatively than the existing physics-based approaches solving conventional optimisation problems with the dynamics equations of motion (see Sec. 4).

3 METHOD

Overview. Our goal is physically-plausible monocular global 3D human motion capture without markers. We follow a learning-based approach trained through a fully-differentiable physics model, see Fig. 2 for an overview. Our framework includes a neural-network-based proportional-derivative (PD) controller that estimates a force vector allowing controlling the kinematic character with dynamics properties to match its pose with the subject's pose in the image sequence. The ground reaction forces are also estimated alongside the 3D motions without requiring any supervisory force annotations. We can also read out and visualise internal and contact forces regressed from the monocular input. Our method accepts sequential 2D joint keypoints in a video (*e.g.*, extracted with an of-the-shelf 2D keypoint detector), and returns 3D skeleton poses which satisfy (bio-)physical constraints. This significantly mitigates foot-floor penetration, body sliding along depth direction and joint jitters. In Sec. 3.1, we define our model and mathematical notations. In Sec. 3.2, we discuss a canonicalisation method of the input 2D joint keypoints which allows our global translation estimation network C_T to be trained jointly on several datasets with different camera intrinsics. In Secs. 3.3 and 3.4, the target pose estimation network and the dynamic cycle with physics-based constraints are elaborated, respectively. In the latter, the 3D pose is updated in the custom optimisation layer where we introduce a hard constraint to prevent foot-floor penetration in a differentiable manner. The obtained 3D poses are smooth, plausible and show mitigated motion delay even on fast motion sequences thanks to the learning-based PD controller which dynamically adjusts the gain parameters depending on the motions in the scene. Our fully-differentiable architecture allows finetuning using 2D annotations only for improved accuracy on in-the-wild footage (Sec. 3.6). Applications of our methods are discussed in Sec. 3.7.

3.1 Our Model, Assumptions and Notations

We represent the kinematic state of the skeleton by a pose vector $\mathbf{q} \in \mathbb{R}^{n+1}$ and its velocity $\dot{\mathbf{q}} \in \mathbb{R}^{n}$ in the camera frame, with $n = 46$. The first seven entries of $\mathbf{q}$ represent the root translation $\mathbf{q}_{\text{trans}} \in \mathbb{R}^3$ and rotation in the quaternion parametrisation $\mathbf{q}_{\text{ori}} \in \mathbb{R}^4$, respectively. All remaining $n - 7$ entries of $\mathbf{q}$ encode joint angles of the human skeleton model parametrised by Euler angles. The first three entries of $\dot{\mathbf{q}}$ represent the linear velocity of the root whereas the next three ones stand for its angular velocity $\omega \in \mathbb{R}^3$. The remaining entries of $\dot{\mathbf{q}}$ stand for the angular velocity of each joint and they correspond to the joint order in $\mathbf{q}$. The time derivative of $\mathbf{q}_{\text{ori}}$ is approximated as follows:

$$\frac{\mathrm{d}\mathbf{q}_{\text{ori}}}{\mathrm{dt}} \approx \frac{1}{2}\begin{bmatrix}0\\ \omega\end{bmatrix} \otimes \mathbf{q}_{\text{ori}}, \tag{1}$$

where ⊗ represents a quaternion multiplication. Eq. (1) is used to update the 3D root orientation from its angular velocity in each dynamics simulation step.

We use M 2D joint keypoints normalised in two different ways, *i.e.*, the root-relative 2D keypoints normalised by the image size and gathered in $\mathbf{K}_{rr} \in \mathbb{R}^{M\times 2}$, and the canonical 2D keypoints stacked in $\mathbf{K}_c \in \mathbb{R}^{M\times 2}$, allowing the network training on datasets with different camera intrinsics. Resorting to root-relative 2D joint keypoints is a widely-used normalisation approach for estimating the root-relative 3D pose from an image or video since it is translation-invariant in the image space. Therefore, we use $\mathbf{K}_{rr}$ for estimating the joint angles and root orientation of the character $\mathbf{q}_{rr}$. On the

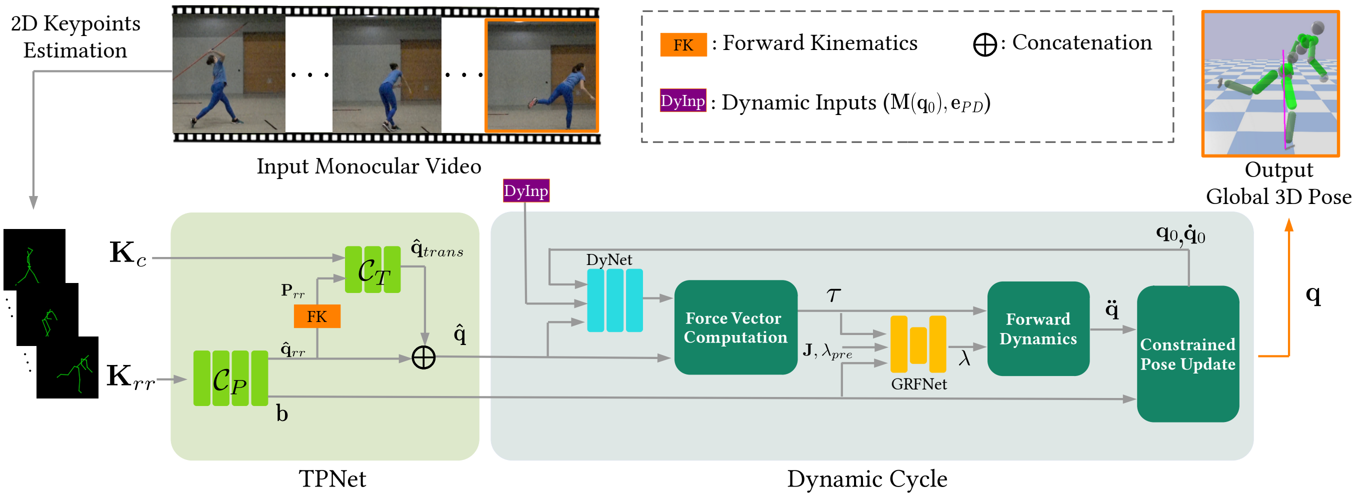

Fig. 2. **Overview of our physionical approach for markerless monocular 3D human motion capture.** Our architecture assumes 2D keypoints in two representations as input, *i.e.*, the canonicalised 2D keypoints ($\mathbf{K}_c$) and root-relative 2D keypoints normalised by the image size ($\mathbf{K}_{rr}$). These representations are complementary and ensure that joint angles and root orientation can be accurately estimated (thanks to $\mathbf{K}_{rr}$) along with the global translation, with no dependency on the camera intrinsics (thanks to $\mathbf{K}_c$). First, the target kinematic pose $\hat{\mathbf{q}}$ is regressed with TPNet and fed to the dynamic cycle which implements various types of physics-based boundary conditions. The dynamic cycle includes several neural components. The input to DyNet is a set of parameters (the target pose $\hat{\mathbf{q}}$, the current pose $\mathbf{q}_0$, the current velocity $\dot{\mathbf{q}}_0$, the mass matrix $\mathbf{M}$ and the current pose error $\mathbf{e}_{\mathrm{PD}} = d(\hat{\mathbf{q}}, \mathbf{q}_0)$) and the outputs are gain parameters k_p of the PD controller and the offset force α for each DoF. The outputs from TPNet and DyNet are used to compute the force vector $\boldsymbol{\tau}$ following the PD controller rule. The GRFNet estimates the ground reaction force $\boldsymbol{\lambda}$. Both $\boldsymbol{\tau}$ and λ are then passed to the forward dynamics module which regresses the accelerations $\ddot{\mathbf{q}}$ in the skeleton frame. This module considers mass matrix of the body $\mathbf{M}$, internal and external forces, gravity, Coriolis and centripetal forces. Finally, the character's pose is updated using the obtained $\ddot{\mathbf{q}}$ through the differentiable optimisation layer to prevent foot-floor penetration.

one hand, this normalisation alone loses the cues for estimating the global translation of the subject in the scene. On the other hand, canonicalised 2D joint keypoints retain the required information to regress the global pose, see Sec. 3.2 for the details.

Our character is composed of *links* which are volumetric body part representations with collision proxies, following the same structure as [Shimada et al. 2020]. Our core idea is to enable awareness of physical laws in our framework which helps to obtain physically-plausible human motion captures. We impose the laws of physics by considering Newtonian rigid body dynamics, which—when applied to our case—reads as [Featherstone 2014]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^{\mathsf{T}}(\mathbf{q})\mathbf{G}\boldsymbol{\lambda} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), \qquad (2)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ are the inertia matrix in the skeleton frame, which describes the moments of inertia of the system, and the acceleration of $\mathbf{q}$, respectively; $\mathbf{J} \in \mathbb{R}^{6N_c \times n}$ is a contact Jacobi matrix which relates velocities in the skeleton frame to velocities in Cartisian coordinates; N_c denotes the number of links to which the contact forces are applied; $\mathbf{G} \in \mathbb{R}^{6N_c \times 3N_c}$ is the matrix that converts the contact force $\boldsymbol{\lambda} \in \mathbb{R}^{3N_c}$ to linear forces and torques (for its details, readers are referred to [Featherstone 2014]); $\mathbf{h} \in \mathbb{R}^n$ encompasses gravity, Coriolis and centripetal forces; the force vector $\boldsymbol{\tau} \in \mathbb{R}^n$ represents the internal joint forces of the character, with its first six entries being the direct root actuations which are set to 0 as per convention.

The total forces that explain the root motion include external forces such as ground reaction force (GRF). Similar to several prior works [Andrews et al. 2016; Levine and Popović 2012; Shimada et al. 2020; Yuan and Kitani 2020], we minimise the direct (virtual) root actuation by estimating the acting GRF and explaining the observed motions with it as much as possible (instead of setting the first six entries of $\boldsymbol{\tau}$ to zero).

3.2 Input Canonicalisation

For the networks that estimate the character's pose without global translation $\mathbf{q}_{rr}$ (*e.g.*, $\mathcal{C}_P$), we use root-relative 2D joint keypoints $\mathbf{K}_{rr}$. Many algorithms, which use a perspective camera model, estimate the global root position by optimising a 2D projection-based loss without learning components [Habermann et al. 2020; Mehta et al. 2020, 2017b]. [Pavllo et al. 2019] and [Shi et al. 2020] employ neural networks to directly regress the translation of the 3D poses. However, in this case the learned motion manifolds depend on the camera intrinsic parameters used during the training. Consequently, at test time, they expect similar camera intrinsics. To tackle this issue, we propose to use canonicalised 2D keypoints $\mathbf{K}_c$ to factor out the influence of the camera intrinsics before they are fed to the neural network that regresses the absolute root translation of the character. Our architecture benefits from the canonicalisation in two ways. First, the translation estimation network can be trained with a large scale joint dataset, *i.e.*, a composition of Human 3.6M [Ionescu et al. 2013], MPI-INF-3D-HP [Mehta et al. 2017a] and DeepCap [Habermann et al. 2020], which are recorded with different intrinsic camera parameters. Second, arbitrary camera intrinsics can be used at test time without influencing the performance of the network that regresses the global translations.

Consider the perspective camera projection without a skew parameter:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{f_x X}{Z} + c_x \\ \frac{f_y Y}{Z} + c_y \\ 1 \end{bmatrix}, \tag{3}$$

where $[X, Y, Z]^{\mathsf{T}}$ is a 3D coordinate of a joint in the camera frame, f the focal length and c the principal point. We see that the 2D joint keypoints $\left[\frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y\right]^{\mathsf{T}}$ are influenced by the camera intrinsic parameters. Therefore, we generate canonical 2D joint keypoints by applying the identity as an intrinsic camera matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix}. \tag{4}$$

We use $[X/Z, Y/Z]^{\mathsf{T}}$ as the canonical 2D joint keypoint which is not influenced by camera intrinsic parameters. In the case, when the depth information Z is not known (*e.g.*, during the testing phase), we can still obtain the canonical 2D joint keypoints assuming that camera intrinsics are known. Let $p_m = [u_m, v_m]^{\mathsf{T}}$ be the 2D joint locations of M joints in the pixel coordinates, with $m \in \{1, \ldots, M\}$. We next stack the canonicalised 2D keypoints in a single $\mathbf{K}_c$ matrix:

$$\mathbf{K}_c = \begin{bmatrix} \frac{u_1 - c_x}{f_x} & \frac{u_2 - c_x}{f_x} & & \frac{u_M - c_x}{f_x} \\ \frac{v_1 - c_y}{f_y} & \frac{v_2 - c_y}{f_y} & \cdots & \frac{v_M - c_y}{f_y} \end{bmatrix}^{\mathsf{T}}. \tag{5}$$

It follows from Eqs. (4) and (5), that for a single p_m and for the corresponding 3D joint location $P_m = [X_m, Y_m, Z_m]^{\mathsf{T}}$, we have:

$$\left[\frac{u_m - c_x}{f_x}, \frac{v_m - c_y}{f_y}\right]^{\mathsf{T}} = \left[\frac{X_m}{Z_m}, \frac{Y_m}{Z_m}\right]^{\mathsf{T}}. \tag{6}$$

This can be interpreted as a point lying on the plane with $Z = 1$. The generalisability and accuracy of the networks trained with the canonicalised 2D keypoints are evaluated in Sec.4.

3.3 Target Pose Estimation

After pre-processing the 2D joint keypoints (Sec. 3.2), the inputs are fed to the target pose estimation network (TPNet) that outputs the global target pose $\hat{\mathbf{q}} \in \mathbb{R}^{n+1}$ and binary labels for the contact states, *i.e.*, toes and heels $\mathbf{b} \in \{0, 1\}^4$, see Fig. 2 for the overview. TPNet consists of two 1D convolution-based network modules (C_T and C_P) that consider temporal information. Network C_P first estimates the joint angles and global orientation of the character without the root translation, which is denoted by $\hat{\mathbf{q}}_{rr}$, and foot contact labels $\mathbf{b}$ in the scene; $\hat{\mathbf{q}}_{rr}$ is further processed by the forward kinematics layer $f(\cdot)$ to obtain the root-relative 3D joint keypoints $\mathbf{P}_{rr}$ with bone lengths in Cartesian coordinates in the absolute scale. Network C_T takes as input $\mathbf{P}_{rr}$ and $\mathbf{K}_c$, and outputs the global translation of the character $\hat{\mathbf{q}}_{\text{trans}}$. At the end, we obtain global 3D skeleton pose $\hat{\mathbf{q}}$ which is further employed as a target pose of the PD controller (Sec. 3.4.1). All the networks in TPNet are composed of four residual blocks with 1D convolution layers with window size 10. Note that our networks accept only past and current frames with no access to future frames, hence compatible with real-time applications.

3.4 Dynamic Cycle

In this section, we elaborate the dynamic cycle of our framework where we control the human character considering dynamics quantities: $\mathbf{M}$, $\mathbf{J}$ and $\mathbf{h}$ are analytically estimated in each simulation step using the current pose $\mathbf{q}_0$ and the velocity $\dot{\mathbf{q}}_0$ [Featherstone 2014].

3.4.1 Force Vector Computation by a Neural PD Controller. PD controllers enable motion tracking with a kinematic character while maintaining a smooth motion. They are hence widely used in robotics and physics-based animation research [Chentanez et al. 2018; Lee et al. 2019; Levine and Popović 2012; Putri et al. 2018; Sugihara and Nakamura 2006]. Our framework also utilises a PD controller to compute the internal force vector $\boldsymbol{\tau}$ of the character. However, the smoothing properties of PD controller can cause motion delay in the presence of fast motions if the gain values are not optimal. The motion delay is especially apparent when the results are shown reprojected to the input views. This issue arises from fixing the gains which adjust the PD controller's sensitivity to the pose and velocity error [Shimada et al. 2020].

Similarly to Chentanez et al. [2018], we dynamically change the gain coefficients depending on the target and current skeleton poses by our dynamics network (DyNet). The latter significantly mitigates the motion delay compared to the existing methods while keeping the motions smooth.

Our DyNet accepts the target pose $\hat{\mathbf{q}}$, the current pose $\mathbf{q}_0$, the current velocity $\dot{\mathbf{q}}_0$, the mass matrix $\mathbf{M}$ and the current pose error $\mathbf{e}_{\text{PD}} = d(\hat{\mathbf{q}}, \mathbf{q}_0) \in \mathbb{R}^n$, and outputs gain parameters $\mathbf{k}_p \in \mathbb{R}^n$ of the PD controller along with the offset forces $\alpha \in \mathbb{R}^n$ for each DoF. The error function $d(\cdot)$ computes entry-wise differences between $\hat{\mathbf{q}}$ and $\mathbf{q}_0$ for the entries that represent the root orientation, we compute the quaternion difference. Since we provide $\hat{\mathbf{q}}$ and $\mathbf{q}_0$, their residual information, *i.e.*, $\mathbf{e}_{\text{PD}}$, is not the essential input for the network. However, similar to [Bergamin et al. 2019], we observed that explicitly providing the current error to DyNet leads to a much faster loss convergence. Therefore, we include $\mathbf{e}_{\text{PD}}$ as one of the inputs to DyNet. The outputs of TPNet and DyNet are used to compute the force vector $\boldsymbol{\tau}$ following the PD controller rule with the compensation term $\mathbf{h}$[1].:

$$\boldsymbol{\tau} = \mathbf{k}_p \circ (\hat{\mathbf{q}} - \mathbf{q}_0) - \mathbf{k}_d \circ \dot{\mathbf{q}}_0 + \alpha + \mathbf{h}, \tag{7}$$

where "∘" denotes Hadamard matrix product. $\mathbf{h}$ represents the sum of gravity, centripetal and Coriolis forces, which are analytically computed.

3.4.2 Ground Reaction Force Estimation. In real world, external forces are required to control the center of gravity of a human body. In other words, for the motion to be physical, the global translation and rotation of the character need to be controlled by external forces such as ground reaction forces obtained from the contact positions. On the other hand, the character motion can be controlled to match the pose of the subject in the scene using the force vector $\boldsymbol{\tau}$. However, $\boldsymbol{\tau}$ contains direct linear and rotational force applied on the root position as elaborated in Sec. 3.4.1.

We thus train the ground reaction force estimation network (GRFNet) to minimise the (virtual) force applied directly on the

[1] In literature, this is known as PD controller with force compensation [Yang et al. 2010].

root, trying to explain the global motion by the ground reaction force $\boldsymbol{\lambda}$ as much as possible. Let $\boldsymbol{\tau}_{\text{root}} \in \mathbb{R}^6$ be the force vector corresponding to the root position (*i.e.*, the first six elements of $\boldsymbol{\tau}$).

Then, the main objective function for training GRFNet reads:

$$\mathcal{L}_{\text{force}} = \left\| \boldsymbol{\tau}_{\text{root}} - \mathbf{J}_1^{\mathsf{T}} \mathbf{G} \boldsymbol{\lambda} \right\|_2^2, \tag{8}$$

where $\mathbf{J}_1^{\mathsf{T}}$ denotes the first six rows of $\mathbf{J}^{\mathsf{T}}$ corresponding to the root configuration. Minimising (8) encourages the network to estimate $\boldsymbol{\lambda}$ which explains the forces applied on the root position by GRF.

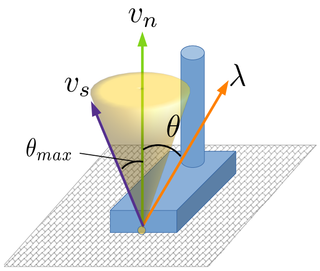

Fig. 3. Schematic visualisation of the friction cone and the ground reaction force at the foot-floor contact position.

However, the direction of the contact force does not only depend on (8). Therefore, we also introduce the friction constraint $\mathcal{F}$ for $\boldsymbol{\lambda}$ to be physically plausible. The estimated $\boldsymbol{\lambda}$ needs to be inside of the so-called friction cone which is defined by the friction coefficient $\mu = 0.8$ together with the normal and tangential directions of the contact position. The friction-cone constraint is defined as follows:

$$\mathcal{F}^{\ell} = \left\{ \boldsymbol{\lambda}^{\ell} \in \mathbb{R}^3 \,\middle|\, \lambda_n^{\ell} > 0, \left\| \lambda_t^{\ell} \right\|_2 \leq \mu \lambda_n^{\ell} \right\}, \tag{9}$$

where ℓ represents the identifier of the link where contact force is applied; λ_n and λ_t represent the normal and tangential component of λ, respectively. We next extend Eq. (9) by integrating the objective function of GRFNet (8) in it:

$$\mathcal{L}_{\text{cone}} = \begin{cases} \|\theta\|_2^2, & \text{if } \theta > \theta_{\max}, \\ 0, & \text{else}, \end{cases} \tag{10}$$

where $\theta_{\max}$ is the angle between the normal vector v_n of the contact position and a vector v_s that lies on the surface of the friction cone, and θ is the angle between v_n and $\boldsymbol{\lambda}$, see Fig. 3. Next, we introduce a temporal smoothness regulariser for the ground reaction force $\boldsymbol{\lambda}$:

$$\mathcal{L}_{\text{smooth}} = \left\| \boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{pre}} \right\|_2^2, \tag{11}$$

where $\boldsymbol{\lambda}_{\text{pre}}$ represents the estimated $\boldsymbol{\lambda}$ in the previous simulation step. The final objective function for GRFNet $\mathcal{L}_{\text{GRF}}$ is as follows:

$$\mathcal{L}_{\text{GRF}} = \mathcal{L}_{\text{force}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{cone}}. \tag{12}$$

3.4.3 Forward Dynamics. To introduce the laws of physics in our 3D motion capture algorithm, we embed the forward dynamics layer in our architecture. We derive joint accelerations $\ddot{\mathbf{q}}$ from Eq. (2):

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})\left(\boldsymbol{\tau}^* + \mathbf{J}^{\mathsf{T}} \mathbf{G} \boldsymbol{\lambda} - \mathbf{h}\right), \tag{13}$$

where $\boldsymbol{\tau}^* = \boldsymbol{\tau} - \mathbf{J}^{\mathsf{T}} \mathbf{G} \boldsymbol{\lambda}$. In this formulation, $\boldsymbol{\tau}^*$ expresses the minimised direct root actuation with contact force compensation for each joint torque. This forward dynamics layer returns $\ddot{\mathbf{q}}$ considering mass matrix of the body $\mathbf{M}$, internal and external forces, gravity, Coriolis and centripetal forces encompassed in $\mathbf{h}$.

3.4.4 Constrained Pose Update. In this step, we update the character's pose using the estimated accelerations $\ddot{\mathbf{q}}$ through the differentiable optimisation layer to prevent foot-floor penetration. Given $\ddot{\mathbf{q}}$ in the skeleton frame and the simulation time step Δt, the velocity $\dot{\mathbf{q}}$ and the kinematic 3D pose $\mathbf{q}$ are updated using the finite differences:

$$\begin{aligned} \dot{\mathbf{q}}^{i+1} &= \dot{\mathbf{q}}^{i} + \Delta t \, \ddot{\mathbf{q}}^{i}, \\ \mathbf{q}^{i+1} &= \mathbf{q}^{i} + \Delta t \, \dot{\mathbf{q}}^{i+1}, \end{aligned} \tag{14}$$

where i denotes the simulation step identifier. To prevent foot-floor penetration, we introduce the differentiable optimisation layer following the formulation of [Agrawal et al. 2019a]. This custom neural network layer solves a specific optimisation problem for each forward pass and returns its derivatives for each backward pass. More specifically, we update the velocity in the skeleton frame $\dot{\mathbf{q}}$ solving the optimisation below:

$$\min_{\dot{\mathbf{q}}^*} \|\dot{\mathbf{q}}^* - \dot{\mathbf{q}}\|, \text{ s.t. } \mathbf{r}_n^c > 0, \tag{15}$$

where $\mathbf{r}_n^c$ represents the linear velocity of the contact position along the normal direction of the floor. Velocity vector $\mathbf{r}^c$ is computed as follows:

$$\mathbf{r}^c = \mathbf{T}(\mathbf{J}\dot{\mathbf{q}}), \tag{16}$$

where $\mathbf{T}(\cdot)$ is the transformation from the camera frame to the floor frame of reference. After solving (15), the estimated $\dot{\mathbf{q}}^*$ is substituted as $\dot{\mathbf{q}}$ in Eq. (14). The dynamic cycle introduced in this section is iterated $k = 6$ times. After the iterations are complete, we obtain the final physically-plausible 3D character's pose $\mathbf{q}$.

3.5 Network Training

We pre-train TPNet for a more stable training of the whole architecture. Such pre-training is advantageous due to two reasons. First, estimating joint angles from 2D joint keypoints leads to ambiguities in bone orientations [Shi et al. 2020]. Second, controlling the dynamic character in 3D by estimated forces to match the subject's pose only from 2D joint keypoints is an ill-posed problem. The network C_P in TPNet is pre-trained with the following objective loss function:

$$\mathcal{L}_{C_P} = \mathcal{L}_{\text{3D}}(\hat{\mathbf{q}}) + \mathcal{L}_{\text{2D}}(\hat{\mathbf{q}}) + \mathcal{L}_{\text{ori}}(\hat{\mathbf{q}}_{\text{ori}}) + \mathcal{L}_{\text{irr.}}(\hat{\mathbf{q}}) + \mathcal{L}_b(\mathbf{b}). \tag{17}$$

The main 3D loss $\mathcal{L}_{\text{3D}}$ is defined as follows:

$$\mathcal{L}_{\text{3D}}(\hat{\mathbf{q}}) = \left\| f(\hat{\mathbf{q}}) - \mathbf{p}'_{\text{3D}} \right\|_2^2, \tag{18}$$

where $f(\cdot)$ and $\mathbf{p}'_{\text{3D}}$ are forward kinematics function and ground-truth 3D joint keypoints, respectively. The loss $\mathcal{L}_{\text{2D}}$ stands for the 2D reprojection error

$$\mathcal{L}_{\text{2D}}(\hat{\mathbf{q}}) = \left\| \Pi(f(\hat{\mathbf{q}})) - \mathbf{p}'_{\text{2D}} \right\|_2^2, \tag{19}$$

where $\Pi(\cdot)$ and $\mathbf{p}'_{\text{2D}}$ are the perspective projection operator and ground-truth 2D joint keypoints normalised by the image size, respectively. The loss $\mathcal{L}_{\text{ori}}$ is added for the supervision of the global root orientation represented by a quaternion:

$$\mathcal{L}_{\text{ori}}(\hat{\mathbf{q}}_{\text{ori}}) = \left\| \hat{\mathbf{q}}_{\text{ori}} \ominus \mathbf{q}'_{\text{ori}} \right\|_2^2, \tag{20}$$

where $\mathbf{q}'_{\text{ori}}$ is the ground-truth root orientation in quaternion parametrisation, and "$\ominus$" denotes a difference computation after converting the quaternion into a rotation matrix. The loss $\mathcal{L}_{\text{irr.}}$ keeps the estimated joint angles in a reasonable range:

$$\mathcal{L}_{\text{irr.}}(\hat{\mathbf{q}}) = \sum_{i=1}^{40} \Psi(\hat{\mathbf{q}}_i), \text{ with} \tag{21}$$

$$\Psi(\hat{\mathbf{q}}_i) = \begin{cases} (\hat{\mathbf{q}}_i - \psi_{\max,i})^2, & \text{if } \hat{\mathbf{q}}_i > \psi_{\max,i} \\ (\psi_{\min,i} - \hat{\mathbf{q}}_i)^2 , & \text{if } \hat{\mathbf{q}}_i < \psi_{\min,i}, \\ 0 \qquad\qquad\quad , & \text{otherwise,} \end{cases} \tag{22}$$

where $\hat{\mathbf{q}}_i$ denotes the joint angle of the i-th joint and $[\psi_{\min,i}, \psi_{\max,i}]$ defines the reasonable angle range for the i-th joint. Term $\mathcal{L}_b$ is the binary cross entropy loss to train the network for estimating correct foot contact states in the scene:

$$\mathcal{L}_b(\mathbf{b}) = -\sum_{i=1}^{4} b'_i \log(b_i) + (1 - b'_i)\log(1 - b_i), \tag{23}$$

where b'_i and b_i are the ground-truth contact label and predicted contact probability on i-th joint, respectively.

The C_T module of TPNet is trained with the 3D translation loss:

$$\mathcal{L}_{C_T}(\hat{\mathbf{q}}_{\text{trans}}) = \left\|\hat{\mathbf{q}}_{\text{trans}} - \mathbf{q}'_{\text{trans}}\right\|_2^2, \tag{24}$$

where $\mathbf{q}'_{\text{trans}}$ denotes the ground-truth translation in 3D space.

After pre-training C_P and C_T with $\mathcal{L}_{C_T}$ and $\mathcal{L}_{C_P}$, we train DyNet with the following loss:

$$\mathcal{L}_{\text{Dyn}}(\mathbf{q}) = \|\mathbf{q} - \hat{\mathbf{q}}\|_2^2 + \varphi \, \|\tau\|_2^2, \tag{25}$$

where $\mathbf{q}$ is the final, physically-plausible 3D pose passed through the differentiable physics model and $\varphi = 10^{-6}$ is the weight of the regularisation term of $\boldsymbol{\tau}$. The first term of $\mathcal{L}_{\text{Dyn}}$ enforces the character to catch up with the target pose with the mitigated motion delay by dynamically estimating the gain parameters of the PD controller. The second term of $\mathcal{L}_{\text{Dyn}}$ prevents overshooting of the PD controller output. The GRFNet is trained with Eq. (12) (Sec. 3.4.2). After pre-training all the networks until convergence, all the networks are trained jointly with the corresponding objective functions with an early stopping strategy. We use Adam optimiser with a learning rate 3.0×10^{-6} for the pre-training, and 3.0×10^{-7} for the joint training.

3.6 Adaptations for In-the-Wild Recordings

Our framework allows finetuning the networks with 2D annotations only using the 2D reprojection loss. Such adjustment of the network weights is especially effective for in-the-wild recordings which differ from the training samples in many aspects (*e.g.*, in the background, lighting conditions or camera poses). We use the estimated 2D joint keypoints from OpenPose [Cao et al. 2019; Cao et al. 2017; Simon et al. 2017; Wei et al. 2016] as pseudo ground-truth 2D annotation to train our network, see Fig. 6 for the results of the ablative study and our supplementary video for visual comparisons of the results with and without finetuning.

3.7 Applications

Since our framework estimates the global translation, root orientation and joint angles, virtual characters can be directly animated using the output of our method. We can also visualise the estimated torques and ground reaction forces that explain the motion in the scene, see Fig. 1-(right) for an example. The purple vectors represent the estimated ground reaction forces, and more saturated green hue on the links represents stronger torques applied on the child joints.

4 EXPERIMENTS

We evaluate our physionical approach for monocular 3D human motion capture on Human 3.6M [Ionescu et al. 2013][2], MPI-INF-3DHP [Mehta et al. 2017a], DeepCap [Habermann et al. 2020] as well as newly recorded sequences. We first provide implementation details (Sec. 4.1) and then show qualitative results (Sec. 4.3) as well as the quantitative outcomes (Sec. 4.3).

4.1 Implementation

Our neural networks are implemented using PyTorch [Paszke et al. 2019] and Python 3.7. Adam optimiser was used to train them. For the computation of dynamics quantities, we use *Rigid Body Dynamics Library* [Felis 2017]. For the implementation of the differentiable optimisation layer we use [Agrawal et al. 2019b], and *Pybullet* [Coumans and Bai 2016] for visualisation purposes. Our approach is evaluated on a workstation with 32 GB RAM, AMD EPYC 7502P 32-Core Processor and NVIDIA QUADRO RTX 8000.

4.2 Network Details

Our implementation of C_T and C_P is composed of 4 1D-convolution-based residual blocks which consider temporal information. GRFNet consists of 4 fully-connected layers with ReLU activation functions excepting the output layer. DyNet forms a two-headed network to estimate the gain parameters of PD controller and offset forces. Two fully-connected layers and ReLU activation functions are used for its hidden layers. One fully-connected layer is used for its output layer followed by Sigmoid and Tanh activation functions for each head of the network. See Appendix A for the network details.

4.3 Quantitative Results

In this section, we compare our method with other related kinematic-based methods, *i.e.*, VNect [Mehta et al. 2017b], HMR [Kanazawa et al. 2018], HMMR [Kanazawa et al. 2019], VIBE [Kocabas et al. 2020a] and MotioNet [Shi et al. 2020], as well as the recent physics-based method PhysCap [Shimada et al. 2020] on benchmark datasets [Habermann et al. 2020; Ionescu et al. 2013; Mehta et al. 2017a].

We follow the evaluation methodology proposed in [Shimada et al. 2020] which suggests comparisons of monocular 3D human motion capture using an extended set of metrics. Along with the standard root-relative 3D joint position accuracy metrics, *i.e.*, mean per-joint position error (MPJPE) [mm] (the lower the better), percentage of correct keypoints [%] and area under ROC curve (AUC) [%] (the higher the better), we report the global 3D translation error and 2D re-projection errors by projecting the estimated 3D joints onto the input and evaluation (unseen) views. Reprojection to evaluation views reveals various effects (related to physical implausibility) which are difficult to access based only on root-relative 3D errors and reprojections to the input views. Further complementary metrics measuring the degree of plausibility of the reconstructed poses are Mean Penetration Error (MPE), Percentage of Non-Penetration (PNP) and temporal consistency error. MPE evaluates the average

[2] All experiments and training using Human 3.6M were conducted at MPII.

Table 1. Comparisons of 3D joint position errors on DeepCap [Habermann et al. 2020], Human 3.6M [Ionescu et al. 2013] and MPI-INF-3DHP[Mehta et al. 2017a] datasets. From the kinematic-based algorithm class, we compare with VNect [Mehta et al. 2017b], HMR [Kanazawa et al. 2018], HMMR [Kanazawa et al. 2019], VIBE [Kocabas et al. 2020b] and MotioNet [Shi et al. 2020]. From the physics-based algorithm class, we compare our method with PhysCap [Shimada et al. 2020]. "†" denotes physics-based algorithms, otherwise a kinematic algorithm. "∗" denotes MotioNet with causal convolutions which does not have access to the future frames, *i.e.*, the similar problem set as our approach. Our approach shows competitive results with kinematic approaches, and outperforms physics-based approaches with a big margin in most scenarios. For DeepCap dataset, the numbers on left and right of our approach represent the 3D accuracy with and without training on DeepCap dataset, respectively.

		DeepCap			Human 3.6M			MPI-INF-3DHP		
		MPJPE [mm]↓	PCK [%]↑	AUC [%]↑	MPJPE [mm]↓	PCK [%]↑	AUC [%]↑	MPJPE [mm]↓	PCK [%]↑	AUC [%]↑
Procrustes	Ours†	**52.6**/63.6	**97.3**/95.9	**67.1**/60.1	58.2	96.1	64.4	99.1	85.5	42.7
	PhysCap†	68.9	95.0	57.9	65.1	94.8	60.6	104.4	83.9	43.1
	MotioNet*	123.0	73.0	31.0	59.1	-	-	-	-	-
	VIBE	80.1	93.3	50.1	**41.5**	-	-	**63.4**	-	-
	VNect	68.4	94.9	58.3	62.7	95.7	61.9	104.5	84.1	43.2
	HMR	77.1	93.8	52.4	54.3	**96.9**	**66.6**	87.8	**87.1**	**50.9**
	HMMR	75.5	93.8	53.1	55.0	96.6	66.2	106.9	79.5	44.8
no Procrustes	Ours†	**72.7**/88.6	**92.6**/85.7	**55.3**/47.4	76.5	**89.5**	**55.0**	134.5	69.8	30.2
	PhysCap†	113.0	75.4	39.3	97.4	82.3	46.4	122.9	72.1	35.0
	MotioNet*	257.4	33.0	13.3	-	-	-	-	-	-
	VIBE	96.7	85.9	42.4	**65.9**	-	-	**97.7**	-	-
	VNect	102.4	80.2	42.4	89.6	85.1	49.0	120.2	**74.0**	**36.1**
	HMR	113.4	75.1	39.0	78.9	88.2	54.1	130.5	69.7	35.7
	HMMR	101.4	81.0	42.0	79.4	88.4	53.8	174.8	60.4	30.8

Table 2. Global 3D translation error on DeepCap dataset [Habermann et al. 2020]. Note that our networks are trained on Human3.6M [Ionescu et al. 2013] and MPI-INF-3DHP [Mehta et al. 2017a], and *not* trained on DeepCap dataset [Habermann et al. 2020].

	Ours	Ours w/o C_T module	Ours w/o input cano.	PhysCap	VNect	VIBE
MPJPE [mm]↓	**62.6**	68.7	105.0	110.5	112.6	244.5

distance between the foot and floor when there is actually a foot-floor contact in the scene (lower is better). PNP shows the ratio of no foot penetration into the floor (higher reflects higher degree of physical plausibility).

3D Joint Positions. Table 1 summarises the root-relative 3D joint position errors. The first and second row blocks report the calculations with and without Procrustes alignment, respectively. On Human 3.6M and MPI-INF-3DHP with Procrustes alignment, the accuracy of our method is average among the compared methods. On Human 3.6M, we obtain a slightly lower MPJPE than VNect, MotioNet and PhysCap while HMR, HMMR and VIBE achieve the lowest errors in overall. On MPI-INF-3DHP, the overall tendency is preserved, though in addition we outperform HMMR. On the DeepCap dataset, we report the results of two different variants, *i.e.*, when the networks are trained on DeepCap dataset + Human3.6M + MPI-INF-3DHP (on the left) and Human3.6M + MPI-INF-3DHP without DeepCap dataset (on the right). Even without using DeepCap dataset for training, ours outperforms other tested algorithms. Compared with Human 3.6M and MPI-INF-3DHP, DeepCap dataset contains challenging motions such as dance, walking backwards, jumping and running sequences. Purely kinematic algorithms tend to fail on these challenging motions. In our case, the magnitudes of inaccuracies are regularised within a reasonable range thanks to the explicit physics model, which results in a lower MPJPE.

Most of the competing methods overfit to a single dataset and cannot generalise well to other datasets. Without Procrustes alignment, our approach outperforms all other evaluated methods on DeepCap dataset, and ranks second on Human 3.6M. We consistently outperform the most related methods on DeepCap, Human 3.6M and MPI-INF-3DHP (with Procrustes), which estimate global 3D human poses and can be directly used for virtual character animation. This list also includes the physics-based PhysCap, *i.e.*, the most closely related method to ours. The high accuracy of purely kinematics methods (in Table 1, those are all methods without "†") on Human 3.6M and MPI-INF-3DHP comes at the price of frequent and sudden changes in the 3D joint positions, which result in jitters and other artefacts. See our supplementary video for the qualitative examples.

Note that the obvious artefacts such as jitter and foot-floor penetration are not revealed by these conventional metrics, which suggests that considering those alone is not enough to judge the motion quality: they do not draw the complete picture, especially when having computer graphics applications in mind; hence, we report several additional metrics to provide a more comprehensive assessment of the motions.

Global Translation Errors. We also qualitatively compare the accuracy of the global character's root position (translation) on the DeepCap dataset in Table 2. Note that we train our method only on Human 3.6M and MPI-INF-3DHP datasets in this experiment, which also evaluates the generalisability of the translation estimator C_T trained with the canonical 2D keypoints. We also show our ablated models 1) without the C_T module and 2) without the input canonicalisation, in the third and fourth columns, respectively. In

Table 3. Comparison of temporal smoothness on the DeepCap [Habermann et al. 2020] and Human 3.6M datasets [Ionescu et al. 2013].

		Ours	PhysCap	VNect	HMR	HMMR	VIBE
DeepCap	e_{smooth}	**5.8**	6.3	11.6	11.7	8.1	7.2
	σ_{smooth}	8.1	4.1	8.6	9.0	5.1	10.1
Human 3.6M	e_{smooth}	**4.5**	7.2	11.2	11.2	6.8	-
	σ_{smooth}	6.9	6.9	10.1	12.7	5.9	-

Table 4. 2D projection error of a frontal view (input) and side view (non-input) on DeepCap dataset [Habermann et al. 2020].

	Front View		Side View	
	e_{2D}^{input} [pix]	$\sigma_{2D}^{\text{input}}$	e_{2D}^{side} [pix]	$\sigma_{2D}^{\text{side}}$
Ours	**7.6**	7.5	**11.5**	**13.1**
PhysCap	21.1	6.7	35.5	16.8
VNect	14.3	2.7	37.2	18.1

the third case, instead of using C_T, we estimate the global translation by solving a 2D reprojection-based optimisation with gradient descent, given the estimated root-relative 3D pose and 2D joint keypoints. Without the input keypoint canonicalisation, the performance of our algorithm is significantly decreased compared to our full model. This is because the network overfits to the camera parameters which are observed in the training datasets without the input canonicalisation. For VIBE—since it does not return a global 3D translation—we apply re-scaling of bone lengths to match the ground-truth bone lengths and likewise solve a reprojection-based optimisation to estimate the global translation which we report in the seventh column. We see that even without C_T module activated, our method outperforms PhysCap, VNect and VIBE by 75% (for PhysCap) and more (VNect and VIBE). See our supplementary video for the qualitative comparisons.

Physical Plausibility Measurement. We further evaluate our approach using quantitative measures for the plausibility of the 3D motion. Table 3 shows the temporal smoothness error e_{smooth} which is computed as follows [Shimada et al. 2020]:

$$e_{\text{smooth}} = \frac{1}{Tk}\sum_{t=1}^{T}\sum_{s=1}^{k} \|\text{Jit}_{\text{GT}} - \text{Jit}_X|, \qquad (26)$$
$$\text{with } \text{Jit}_X = \left\|\mathbf{p}_X^{s,t} - \mathbf{p}_X^{s,t-1}\right\| \text{ and } \text{Jit}_{\text{GT}} = \left\|\mathbf{p}_{\text{GT}}^{s,t} - \mathbf{p}_{\text{GT}}^{s,t-1}\right\|,$$

where $\mathbf{p}^{s,t}$ represents the 3D position of joint s in the frame t; T and k denote the total numbers of frames in the input sequence and target 3D joints, respectively. Smaller e_{smooth} means less jitter in the reconstructed 3D motions. Our approach shows the lowest e_{smooth} among all tested methods, followed by physics-based method PhysCap and VIBE and HMMR with temporal constraints (*i.e.*, these methods take several frames as inputs). This confirms the significance of our explicit physics model for more physically-plausible results.

We also report in Table 4 the 2D reprojection error onto the input views (e_{2D}^{input}) and side views (e_{2D}^{side}) that are not used as inputs to the algorithms: $\sigma_{2D}^{\text{input}}$ and $\sigma_{2D}^{\text{side}}$ represent the standard deviation of e_{2D}^{input} and e_{2D}^{side}, respectively. Reprojection onto non-input-view is an expressive operation, since it reveals the artefacts which are not observable from the input view (*e.g.*, body leaning and wrong translation estimation along the depth direction). Again, our results lead to the lowest metric among all methods which suggests that our global 3D motion capture is more physically plausible compared to other methods.

Table 5. Comparison of Mean Penetration Error (MPE) and Percentage of Non-Penetration (PNP) on DeepCap dataset [Habermann et al. 2020].

	MPE [mm]↓	PNP [%]↑
Ours	28.9	92.3
Ours w/o HC	29.7	89.6
PhysCap	**28.0**	**92.9**
VNect	39.3	45.6

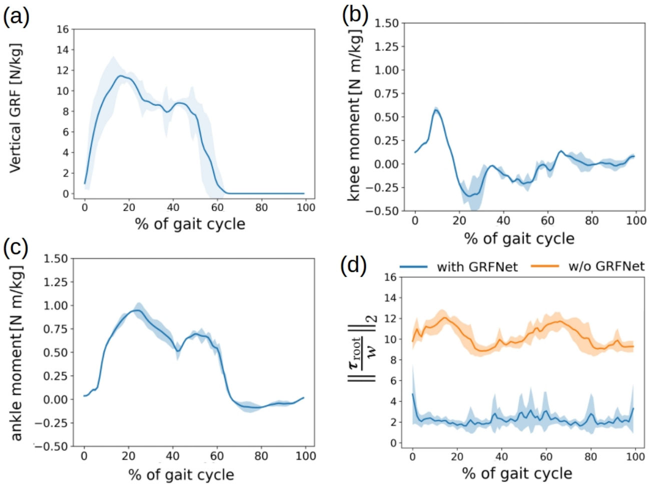

Fig. 4. Estimated forces of the walking sequences from the DeepCap dataset. The thick line and coloured area represent the means and standard deviations, respectively. The force graph lies in the reasonable range for walking motion (*cf.* [Shahabpoor and Pavic 2017; Zell et al. 2020]), and mostly shows a smooth curve.

Finally, Table 5 reports the physical plausibility measurement for foot-floor penetration. Our result is on par with PhysCap which introduces hard constraint to prevent foot-floor penetration, followed by the purely kinematic method VNect. We also show our ablated model without the hard constraint layer (Sec. 3.4.4). Compared to it, our full architecture shows better performance in terms of the foot-floor penetration metric.

GRF Function. In Fig. 4, we plot the forces estimated by our physionical algorithm for the walking motion from the DeepCap dataset. The thick lines and coloured areas represent the mean values and standard deviations, respectively. In Figs. 4-(a), (b) and (c), we show the estimated GRF along the vertical direction and joint torques of knee and ankle, respectively. The curve is smooth and is in a reasonable range for walking motions. Interested readers are referred

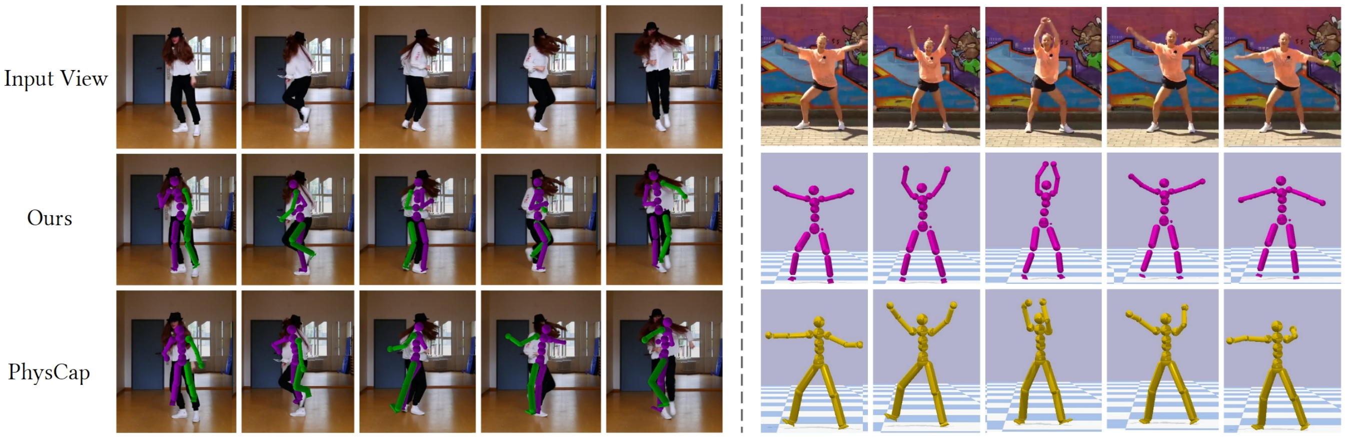

Fig. 5. Qualitative comparisons of methods with physics-based constraints on videos with fast motions. While having a consistently improved accuracy on general motions compared to PhysCap, our approach can capture significantly faster motions as it learns motion priors and the associated gains of the neural PD controller from data.

to [Shahabpoor and Pavic 2017; Zell et al. 2020] for a visual comparison with ground-truth GRF curves for an exemplary walking sequence obtained with force plates. Note that our approach accepts only a single 2D image sequence as input and does not require any ground-truth forces for its training unlike [Zell et al. 2020]. In Fig. 4-(d), we show an ablative study of GRFNet. As elaborated in Sec. 3.4.2, GRFNet minimises the presence of unnatural virtual forces directly applied on the character's root joint $\boldsymbol{\tau}_{\text{root}}$ and tries to explain the root motion by the GRF only, as much as possible. We report $\left\|\frac{\boldsymbol{\tau}_{\text{root}}}{w}\right\|_2$ for walking cycles, where w is the character's weight. Without GRFNet, the magnitude of the virtual force acting directly on the root is ~5 times higher compared to the case with the former. This suggests that GRFNet helps to estimate more physically-plausible forces in the proposed framework.

4.4 Qualitative Results

We further show results on multiple in-the-wild sequences. All in all, we observe that our physionical method outputs temporally-consistent global 3D human poses which not only accurately project to the input views but which also look physically plausible when observed from arbitrary views in the 3D space. Our reconstructed 3D motions show significantly mitigated physically-implausible artifacts such as spurious global translational variations along the depth dimension, foot-floor penetration and jitters, see our supplemental video for the qualitative results.

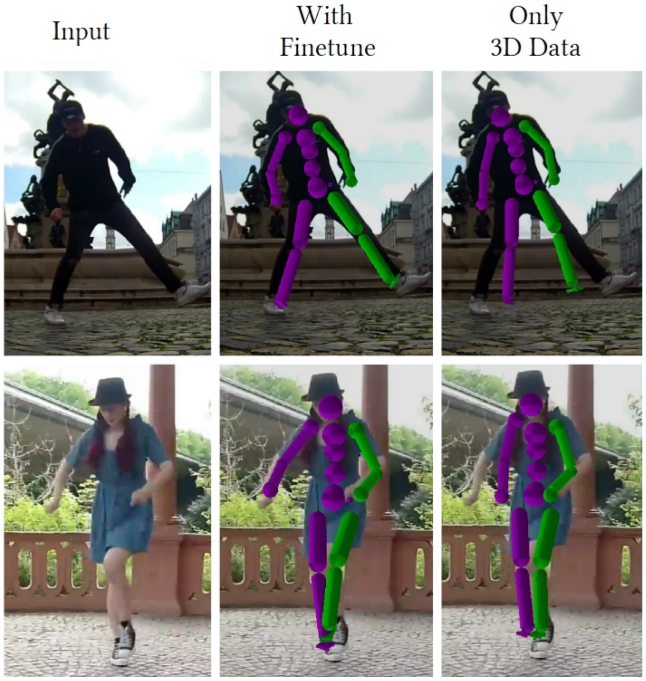

Fig. 6. The accuracy of our method with finetuning using additional 2D annotations improves for in-the-wild sequences, compared to training using 3D data only.

We qualitatively compare our method with the most related work PhysCap [Shimada et al. 2020] in Fig. 5. It is noticeable that our method catches up with fast motions with significantly mitigated motion delay thanks to the learned PD controller gain values for different motion types (see Fig. 5-(left)). PhysCap struggles to reconstruct correct 3D motions when fast motion appears due to its fixed gain parameters of the PD controller. Also note that our framework shows more accurate articulations on the in the-wild-sequence (see Fig. 5-(right)). In Fig. 7, we compare our method with the state-of-the-art kinematic-based methods VNect [Mehta et al. 2017b] and VIBE [Kocabas et al. 2020b] on in-the-wild sequences. Only our method reconstructs smooth sequential 3D motions. The 3D motions by VNect and VIBE show sudden changes in joint positions which are observed as jitters in the video.

We next show the results of our approach with and without finetuning our network with 2D keypoints obtained on the sequences

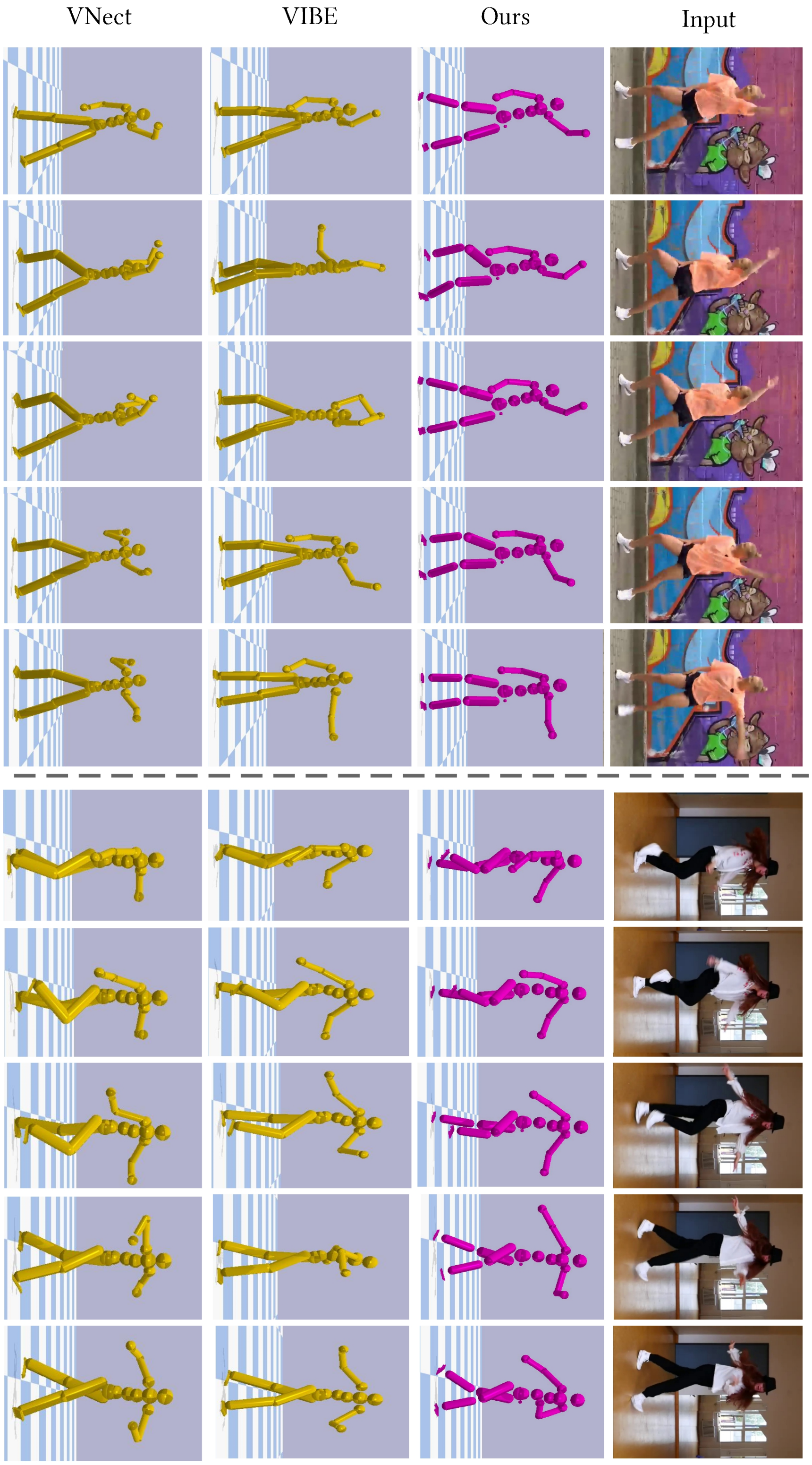

Fig. 7. Results of our method compared to purely-kinematic methods VIBE (3D human pose estimation) [Kocabas et al. 2020b] and VNect (3D human motion capture) [Mehta et al. 2017b]. Our reconstructions are more temporally smooth, whereas the competing methods show frame-to-frame jitter along all axes. See our supplementary video for dynamic visualisations.

in the wild, see Fig. 6 for the qualitative comparison. We use OpenPose [Cao et al. 2019] to obtain 2D keypoints, and the networks are finetuned with the 2D reprojection loss. After the finetuning, our framework shows better overlay and visually more accurate 3D motions compared to the networks trained with the 3D benchmark datasets only (Human 3.6M, MPI-INF-3DHP and DeepCap).

5 CONCLUSIONS

We introduced a new fully-neural approach for 3D human motion capture from monocular RGB videos with hard physics-based constraints which runs at interactive framerates and achieves state-of-the-art results on multiple metrics. Our neural physical model allows learning motion priors and the associated physical properties, as well as gain values of the neural PD controller from data. Thanks to the custom neural layer, which expresses hard physics-based constraints, our architecture is fully-differentiable. In addition, it can be trained jointly on several datasets thanks to the new form of input canonicalisation. Our experiments demonstrate that compared to PhysCap—a recent method with physics-based boundary conditions—our physionical approach captures significantly faster motions, while being more accurate in terms of various 3D reconstruction metrics. Thanks to the full differentiability, the proposed method can be finetuned on datasets with 2D annotations only, which improves the reconstruction fidelity on in-the-wild footages. These properties make it well suitable for direct virtual character animation from monocular videos, without requiring any further post-processing of the estimated global 3D poses.

We believe that the proposed method opens up multiple directions for future research. Our architecture can be classified as a 2D keypoint lifting approach, which has both advantages (*e.g.*, the possibility of 2D keypoint normalisation, on the one hand) and downsides (*e.g.*, reliance on the accuracy of 2D keypoint detectors, on the other). Next, our results naturally lead to the question of what is the most effective way to integrate physics-based boundary conditions in neural architectures, and how the proposed ideas can be applied to many related problem settings.

ACKNOWLEDGMENTS

All data captures and evaluations were performed at MPII by MPII. The authors from MPII were supported by the ERC Consolidator Grant 4DRepLy (770784). We also acknowledge support from Valeo.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. 2019a. Differentiable convex optimization layers. In *Advances in neural information processing systems (NeurIPS)*.

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. 2019b. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sheldon Andrews, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell. 2016. Real-Time Physics-Based Motion Capture with Sparse Sensors. In *European Conference on Visual Media Production (CVMP)*.

Ronen Barzel, John F. Hughes, and Daniel N. Wood. 1996. Plausible Motion Simulation for Computer Graphics Animation. In *Eurographics Workshop on Computer Animation and Simulation*.

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Transactions on Graphics (TOG)* 38, 6 (2019).

Liefeng Bo and Cristian Sminchisescu. 2008. Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision (IJCV)* 87 (2008), 28–52.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *European Conference on Computer Vision (ECCV)*.

Thomas Brox, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers. 2010. Combined Region and Motion-Based 3D Tracking of Rigid and Articulated Objects. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32, 3 (2010), 402–415.

Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2019).

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Computer Vision and Pattern Recognition (CVPR)*.

Ching-Hang Chen and Deva Ramanan. 2017. 3D Human Pose Estimation = 2D Pose Estimation + Matching. In *Computer Vision and Pattern Recognition (CVPR)*.

Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. 2018. Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*. 1–10.

Erwin Coumans and Yunfei Bai. 2016. Pybullet, a python module for physics simulation for games, robotics and machine learning. *GitHub repository* (2016).

Rishabh Dabral, Nitesh B Gundavarapu, Abhishek Mitra, Rahuland Sharma, Ganesh Ramakrishnan, and Arjun Jain. 2019. Multi-Person 3D Human Pose Estimation from Monocular Images. In *International Conference on 3D Vision (3DV)*.

Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. 2018. Learning 3D Human Pose from Structure and Motion. In *European Conference on Computer Vision (ECCV)*.

Hooman Dejnabadi, Brigitte M. Jolles, Emilio Casanova, Pascal Fua, and Kamiar Aminian. 2006. Estimation and visualization of sagittal kinematics of lower limbs orientation using body-fixed sensors. *Transactions on Biomedical Engineering* 53, 7 (2006), 1385–1393.

Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Micha Andriluka, Chris Bregler, Bernt Schiele, and Christian Theobalt. 2015. Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras. In *Computer Vision and Pattern Recognition (CVPR)*.

Matteo Fabbri, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara. 2020. Compressed Volumetric Heatmaps for Multi-Person 3D Pose Estimation. In *Computer Vision and Pattern Recognition (CVPR)*.

Roy Featherstone. 2014. *Rigid body dynamics algorithms*.

Martin L. Felis. 2017. RBDL: an Efficient Rigid-Body Dynamics Library using Recursive Algorithms. *Autonomous Robots* 41, 2 (2017), 495–511.

Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. 2010. Optimization and Filtering for Human Motion Capture - a Multi-Layer Framework. *International Journal of Computer Vision (IJCV)* 87, 1 (2010), 75–92.

Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2020. DeepCap: Monocular Human Performance Capture Using Weak Supervision. In *Computer Vision and Pattern Recognition (CVPR)*.

Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. 2019. In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations. In *Computer Vision and Pattern Recognition (CVPR)*.

Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. 2019. Resolving 3D Human Pose Ambiguities with 3D Scene Constraints. In *International Conference on Computer Vision (ICCV)*.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 36, 7 (2013), 1325–1339.

Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C. Karen Liu. 2019. Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation. *ACM Transactions On Graphics (TOG)* 38, 4 (2019).

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Recognition (CVPR)*.

Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3D Human Dynamics from Video. In *Computer Vision and Pattern Recognition (CVPR)*.

Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020a. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Computer Vision and Pattern Recognition (CVPR)*.

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. 2020b. VIBE: Video inference for human body pose and shape estimation. In *Computer Vision and Pattern Recognition (CVPR)*.

Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. 2019. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *International Conference on Computer Vision (ICCV)*.

Onorina Kovalenko, Vladislav Golyanik, Jameel Malik, Ahmed Elhayek, and Didier Stricker. 2019. Structure from Articulated Motion: Accurate and Stable Monocular

3D Reconstruction without Training Data. *Sensors* 19, 20 (2019).

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable Muscle-Actuated Human Simulation and Control. *ACM Transactions On Graphics (TOG)* 38, 4 (2019).

Sergey Levine and Jovan Popović. 2012. Physically Plausible Simulation for Character Animation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation.*

Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic. 2019. Estimating 3D Motion and Forces of Person-Object Interactions from Monocular Video. In *Computer Vision and Pattern Recognition (CVPR).*

Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-Based Contact-Rich Motion Control. *ACM Transactions On Graphics (TOG)* 29, 4 (2010), 128:1–128:10.

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. 2018. LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering. *ACM Transactions On Graphics (TOG)* 37, 6 (2018).

Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A Simple Yet Effective Baseline for 3D Human Pose Estimation. In *International Conference on Computer Vision (ICCV).*

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017a. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In *International Conference on 3D Vision (3DV).*

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohammad Elgharib, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera. *ACM Transactions on Graphics (TOG)* 39, 4.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017b. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics* 36, 4, 14.

Francesc Moreno-Noguer. 2017. 3D Human Pose Estimation From a Single Image via Distance Matrix Regression. In *Computer Vision and Pattern Recognition (CVPR).*

Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV).*

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS).*

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Computer Vision and Pattern Recognition (CVPR).*

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. 2018. Learning to estimate 3D human pose and shape from a single color image. In *Computer Vision and Pattern Recognition (CVPR).*

Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Computer Vision and Pattern Recognition (CVPR).*

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018a. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018).

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018b. SFV: Reinforcement Learning of Physical Skills from Videos. *ACM Transactions On Graphics (TOG)* 37, 6 (2018).

Dewi Indriati Hadi Putri, Carmadi Machbub, et al. 2018. Gait Controllers on Humanoid Robot Using Kalman Filter and PD Controller. In *International Conference on Control, Automation, Robotics and Vision (ICARCV).*

Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. 2020. Contact and Human Dynamics from Monocular Video. In *European Conference on Computer Vision (ECCV).*

Helge Rhodin, Mathieu Salzmann, and Pascal Fua. 2018. Unsupervised Geometry-Aware Representation Learning for 3D Human Pose Estimation. In *European Conference on Computer Vision (ECCV).*

Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2019. LCR-Net++: Multi-Person 2D and 3D Pose Detection in Natural Images. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2019).

Erfan Shahabpoor and Aleksandar Pavic. 2017. Measurement of Walking Ground Reactions in Real-Life Environments: A Systematic Review of Techniques and Technologies. *Sensors* 17, 9 (2017), 2085.

Dana Sharon and Michiel van de Panne. 2005. Synthesis of Controllers for Stylized Planar Bipedal Walking. In *International Conference on Robotics and Animation (ICRA).*

Mingyi Shi, Kfir Aberman, Andreas Aristidou, Taku Komura, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020. MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency. *ACM Transactions on Graphics (TOG)* 40, 1 (2020), 1–15.

Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. 2020. PhysCap: physically plausible monocular 3D motion capture in real time. *ACM Transactions on Graphics (TOG)* 39, 6 (2020).

Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *Computer Vision and Pattern Recognition (CVPR).*

Jie Song, Xu Chen, and Otmar Hilliges. 2020. Human Body Model Fitting by Learned Gradient Descent. (2020).

Jonathan Starck and Adrian Hilton. 2007. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications (CGA)* 27, 3 (2007), 21–31.

Tomomichi Sugihara and Yoshihiko Nakamura. 2006. Gravity compensation on humanoid robot control with robust joint servo and non-integrated rate-gyroscope. In *International Conference on Humanoid Robots (ICHR).*

Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, YiLi Fu, and Tao Mei. 2019. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *International Conference on Computer Vision (ICCV).*

Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. 2011. Motion Reconstruction Using Sparse Accelerometer Data. *ACM Trans. Graph.* 30, 3 (2011).

Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2016. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *British Machine Vision Conference (BMVC).*

Denis Tomè, Chris Russell, and Lourdes Agapito. 2017. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. In *Computer Vision and Pattern Recognition (CVPR).*

Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. 2007. Practical Motion Capture in Everyday Surroundings. *ACM Trans. Graph.* 26, 3 (2007).

Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Annual Conference of the European Association for Computer Graphics (Eurographics)* (2017), 349–360.

Marek Vondrak, Leonid Sigal, Jessica Hodgins, and Odest Jenkins. 2012. Video-based 3D Motion Capture Through Biped Control. *ACM Transactions On Graphics (TOG)* 31, 4 (2012).

Bastian Wandt and Bodo Rosenhahn. 2019. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In *Computer Vision and Pattern Recognition (CVPR).*

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Computer Vision and Pattern Recognition (CVPR).*

Xiaolin Wei and Jinxiang Chai. 2010. Videomocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences. *ACM Transactions on Graphics (TOG)* 29, 4 (2010).

Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. 2018. Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1560–1567.

Pawel Wrotek, Odest Chadwicke Jenkins, and Morgan McGuire. 2006. Dynamo: Dynamic, Data-Driven Character Control with Adjustable Balance. In *ACM Sandbox Symposium on Video Games.*

Chenglei Wu, Kiran Varanasi, and Christian Theobalt. 2012. Full Body Performance Capture under Uncontrolled and Varying Illumination: A Shading-Based Approach. In *European Conference on Computer Vision (ECCV).*

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. 2019. Monocular Total Capture: Posing Face, Body, and Hands in the Wild. In *Computer Vision and Pattern Recognition (CVPR).*

Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. 2020. EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera. In *Computer Vision and Pattern Recognition (CVPR).*

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. MonoPerfCap: Human Performance Capture From Monocular Video. *ACM Trans. Graph.* 37, 2 (2018).

Chifu Yang, Qitao Huang, Hongzhou Jiang, O Ogbobe Peter, and Junwei Han. 2010. PD control with gravity compensation for hydraulic 6-DOF parallel manipulator. *Mechanism and Machine theory* 45, 4 (2010), 666–677.

Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 2018. 3D Human Pose Estimation in the Wild by Adversarial Learning. In *Computer Vision and Pattern Recognition (CVPR).*

Ye Yuan and Kris Kitani. 2020. Residual Force Control for Agile Human Behavior Imitation and Extended Motion Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS).*

Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. 2018. Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes - The Importance of Multiple Scene Constraints. In *Computer Vision and Pattern Recognition (CVPR).*

Petrissa Zell, Bodo Rosenhahn, and Bastian Wandt. 2020. Weakly-Supervised Learning of Human Dynamics. In *European Conference on Computer Vision (ECCV)*.

Petrissa Zell, Bastian Wandt, and Bodo Rosenhahn. 2017. Joint 3D Human Motion Capture and Physical Analysis from Monocular Videos. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. 2020. Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild. In *European Conference on Computer Vision (ECCV)*.

Yu Zheng and Katsu Yamane. 2013. Human Motion Tracking Control with Strict Contact Force Constraints for Floating-Base Humanoid Robots. In *International Conference on Humanoid Robots (Humanoids)*.

Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. 2017. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach. In *International Conference on Computer Vision (ICCV)*.

Yuliang Zou, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang. 2020. Reducing Footskate in Human Motion Reconstruction with Ground Contact Constraints. In *Winter Conference on Applications of Computer Vision (WACV)*.

A NETWORK DETAILS

We schematically visualise the network details in Fig. 8. Our implementations of $\mathcal{C}_T$ and $\mathcal{C}_P$ are based on [Zou et al. 2020] and composed of 1D convolutional layers with residual blocks. We use the replication padding layer of size 1 for the embedding block and size 4 for the residual block. The kernel size of the 1D convolutional layer for the embedding and residual blocks are 3 and 5, respectively. For the 1D convolution in the residual blocks, we use the dilation of size 2. For $\mathcal{C}_T$—although it is possible to estimate $\hat{\mathbf{q}}_{rr}$ and $\mathbf{b}$ with a single neural network—we observed that estimating the global rotation, joint angles and contact labels with three different networks shows higher accuracy. Therefore, $\mathcal{C}_P$ consists of three replicated networks with the difference in the output layer, see Fig. 8 for the details. For GRFNet and DyNet, all the inputs are concatenated to one vector and fed to the networks. We can estimate $\mathbf{k}_p$ and α directly by DyNet, however, similar to [Chentanez et al. 2018], we obtain $\mathbf{s}_g$ and $\mathbf{s}_f$ ($0 < \mathbf{s}_g < 1$ and $-1 \leq \mathbf{s}_f \leq 1$) using sigmoid and tanh functions, and compute $\mathbf{k}_p = 2\mathbf{s}_g\mathbf{k}_p^{\text{ini}}$ and $\alpha = \gamma\mathbf{s}_f$; $\mathbf{k}_p^{\text{ini}}$ denotes the initial gain parameters which are determined following [Shimada et al. 2020], and γ is the coefficient which is determined empirically. Note that we use the fixed $\mathbf{k}_p^{\text{ini}}$ and $\gamma = 10$ values through all the experiments. We observed that this formulation leads to an improved stability and faster convergence of the network training than directly estimating $\mathbf{k}_p$ and α, since the network outputs are always within the normalised range.

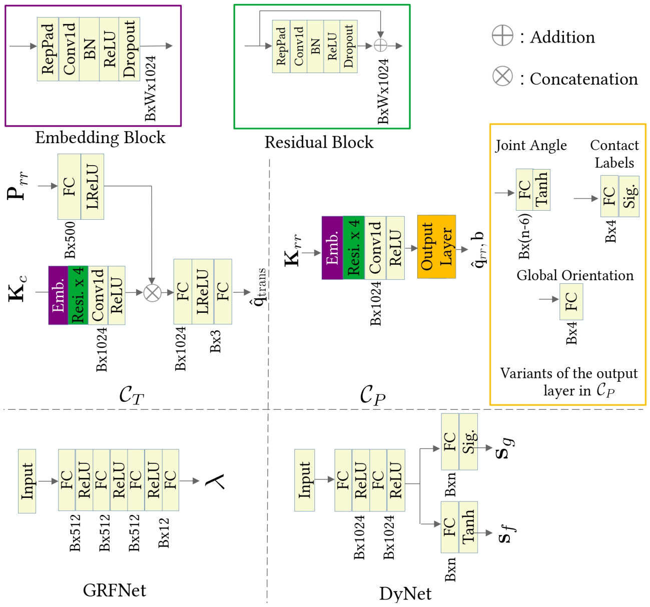

Fig. 8. Schematic visualisations of the network details. “Emb.” and “Resi.” stand for the embedding block (purple box) and residual block (green box), respectively. “BN”, “RepPad”, “FC”, “Sig.” and “Conv1D” represent batch normalisation, replication padding, fully-connected layer, sigmoid function and 1D convolution, respectively. The numbers next to the layers represent the output dimensionality. “B” and “W” represent the batch size and temporal window size, respectively.