Discovering Fatigued Movements for Virtual Character Animation

Noshaba Cheema
noshaba.cheema@dfki.de
DFKI, Max Planck Institute for Informatics, Saarland Informatics Campus
Germany

Rui Xu
rui.xu@dfki.de
DFKI, Saarland Informatics Campus
Germany

Nam Hee Kim
namhee.kim@aalto.fi
Aalto University
Finland

Perttu Hämäläinen
perttu.hamalainen@aalto.fi
Aalto University
Finland

Vladislav Golyanik
golyanik@mpi-inf.mpg.de
Max Planck Institute for Informatics, Saarland Informatics Campus
Germany

Marc Habermann
mhaberma@mpi-inf.mpg.de
Max Planck Institute for Informatics, Saarland Informatics Campus
Germany

Christian Theobalt
theobalt@mpi-inf.mpg.de
Max Planck Institute for Informatics, Saarland Informatics Campus
Germany

Philipp Slusallek
philipp.slusallek@dfki.de
DFKI, Saarland Informatics Campus
Germany

ABSTRACT

Virtual character animation and movement synthesis have advanced rapidly during recent years, especially through a combination of extensive motion capture datasets and machine learning. A remaining challenge is interactively simulating characters that fatigue when performing extended motions, which is indispensable for the realism of generated animations. However, capturing such movements is problematic, as performing movements like backflips with fatigued variations up to exhaustion raises capture cost and risk of injury. Surprisingly, little research has been done on faithful fatigue modeling. To address this, we propose a deep reinforcement learning-based approach, which—for the first time in literature—generates control policies for full-body physically simulated agents aware of cumulative fatigue. For this, we first leverage Generative Adversarial Imitation Learning (GAIL) to learn an expert policy for the skill; Second, we learn a fatigue policy by limiting the generated constant torque bounds based on endurance time to non-linear, state- and time-dependent limits in the joint-actuation space using a Three-Compartment Controller (3CC) model. Our results demonstrate that agents can adapt to different fatigue and rest rates interactively, and discover realistic recovery strategies without the need for any captured data of fatigued movement.

CCS CONCEPTS

• **Computing methodologies → Animation**; **Physical simulation**; *Procedural animation*; *Adversarial learning*; Motion capture.

KEYWORDS

character animation, physics-based animation, reinforcement learning, adversarial learning, cumulative fatigue modeling, biomechanics

ACM Reference Format:
Noshaba Cheema, Rui Xu, Nam Hee Kim, Perttu Hämäläinen, Vladislav Golyanik, Marc Habermann, Christian Theobalt, and Philipp Slusallek. 2023. Discovering Fatigued Movements for Virtual Character Animation. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3610548.3618176

1 INTRODUCTION

For many applications, ranging from games and films to robotics, synthesizing life-like and realistic character animation and control is a crucial yet challenging element. A relatively unexplored yet important area in this direction is the adequate simulation of physiological changes over time. In particular, this is true for cumulative fatigue, i.e., humans getting tired over time when performing strenuous motions. Simulating this effect plausibly aids to the overall realism in interactive applications, such as sports simulators, where athletes start to change their execution patterns due to muscle fatigue, and in games, when virtual characters run out of resources. To achieve this, current games use comparably simple heuristics and pre-defined kinematic animations in state-machines such as motion graphs [Kovar et al. 2002; Lee et al. 2002] to model character fatigue. However, *many* such methods do not incorporate

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12.
https://doi.org/10.1145/3610548.3618176

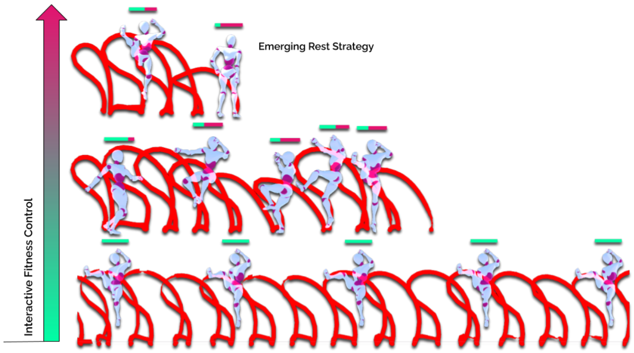

Figure 1: We propose a deep reinforcement learning approach, which explicitly accounts for a character's fatigue.We show that our agents discover novel fatigued motions and recovery strategies without requiring explicit fatigued data. Moreover, our approach enables interactive fitness control within one policy, letting characters fatigue faster or slower depending on the user

any biomechanical realism, require *prohibitively* extensive motion capture data of fatigued motions [Kider Jr et al. 2011].

Previous work targeting character animation can be roughly categorized into data-driven kinematic-based methods (KM) [Büttner and Clavet 2015; Holden et al. 2017; Levine et al. 2012; Min and Chai 2012; Starke et al. 2020] and physics-based simulation, paired with Deep Reinforcement Learning (DRL) [Baker et al. 2019; Jiang et al. 2019; Pathak et al. 2017; Peng et al. 2021; Yin et al. 2021]. KMs provide an easy means to make use of the inherent realism that comes from captured or hand-animated motion data, lacks the ability to generalize to novel behaviours. Meanwhile, characters animated via KMs lack the ability to react to dynamic stimuli such as external perturbations, unless the prior motion capture incorporates the vast amount of interaction and perturbation scenarios. On the other hand, DRL methods provide a promising direction in this field, as even sparse rewards or space constraints already allow for the automatic generation of interactive movements and novel emergent behaviours without the need for explicitly capturing variations of data that fulfill the given constraints. However, what kind of rewards or constraints constitute for realistic movements and behaviours remains a fundamental challenge in animation research. Interestingly, none of the previous works in either of the two categories has focused on modeling character fatigue in a biologically plausible fashion while overall animation realism would benefit from such an approach. Only a few works in the biomechanically-based simulation literature [Cheema et al. 2020; Komura et al. 2000] have looked into this problem. However, previous works have only been applied to carefully hand-crafted single limb settings, as they require expensive musculoskeletal models [Komura et al. 2000] or careful reward modeling and hyper-parameter tuning [Cheema et al. 2020] and are not able to demonstrate the emergence of fatigue and recovery behaviours to the same extent as our work.

In this paper, we propose a novel fatigue-aware policy generation framework for physically simulated characters, which allows for the emergence of realistic fatigue and recovery effects. For this we explore a two-step approach: First, we pre-train the policy to allow for a viable number of stable behaviours to emerge, and a well-behaved initial state for fatigue transfer-learning. Second, based on the learned stable behaviours, we refine the policy by modifying the emerging torque limits to nonlinear, state- and time-dependent limits using a Three-Compartment Controller (3CC) model [Xia and Frey Law 2008], which we adapt from the biomechanics literature in a new context not envisioned by the original paper. To show the effectiveness of our framework, we integrate our fatigue module into a state-of-the-art generative adversarial imitation learning (GAIL) framework [Ho and Ermon 2016; Peng et al. 2021; Torabi et al. 2018].

In summary, the main contributions of this work are as follows:

- **Novel Functionality.** The first approach for emergent fatigue and recovery behaviours in full-body Reinforcement Learning for 3D virtual character animation.
- **Efficient Torque-based Fatigue-System for Animation.** The first full-body fatigue system based solely on joint actuation torques for interactive character simulation.
- **Interactive Fatigue and Rest Control.** The use of a Three-Compartment Controller (3CC) fatigue model to limit joint actuation torques based on observed fatigue and residual strength capacity, which allows for interactive control of different fitness levels within one policy.

Our results demonstrate several emergent fatigue behaviours during repetitive athletic tasks, such as arms bending during cartwheels and decreasing kick and jump height after athletic martial arts kicks (Fig. 1), as well as waiting behaviours to recover from the fatigue to be able to continue. Our agents learn unseen motion patterns while resting in order to most effectively recover from the experienced fatigue. For example, our agent learns to effectively compensate for the momentum after a backflip or a cartwheel while ending up in a motion state where motor units start to recover. As the risk of injury could prohibit motion capture of fatigued yet complex movements, our method brings an added benefit of bypassing this constraint.

2 RELATED WORK

This section reviews physics-based approaches, deep reinforcement learning, muscoskeletal methods and techniques supporting fatigue. Note that as fatigue modeling relies on the physical forces, purely kinematic methods cannot be applied to our problem unless one explicitly captures fatigued motions.

Physics-based Methods. Physics-based methods allow movement generation with physical realism and environmental interaction; they give direct insights on the forces required and being applied to the character for a given task by leveraging a more general knowledge of the physical equations of motion [Geijtenbeek et al. 2013; Raibert and Hodgins 1991; Wampler et al. 2014]. A fundamental challenge in physics-based approaches is the design of controllers for simulated characters. Task-specific controllers (e.g. for locomotion), achieved significant success [Coros et al. 2010; Felis and Mombaur 2016; Geijtenbeek et al. 2013; Lee et al. 2010; Ye and Liu 2010a; Yin et al. 2007]. However, such manually designed controllers remain hard to generalize to diverse movements and tasks. With the wide availability of motion capture data, tracking-based controllers have become a popular research domain [Hämäläinen et al. 2015; Tassa et al. 2012; Wampler et al. 2014; Ye and Liu 2010b], though they remain limited in motion quality and long-term planning. More recently, deep reinforcement learning-based methods have become a promising research direction to account for long-term planning as well as emergent and reactive behaviour – three aspects necessary for emergent fatiguing behaviour over multiple repetitions.

Deep Reinforcement Learning. Deep reinforcement learning (DRL) has been successfully applied to physics-based characters animation [Liu and Hodgins 2017; Peng et al. 2016; Teh et al. 2017]. Here, policy gradient methods emerged for continuous control problems [Schulman et al. 2015, 2017; Sutton et al. 1998]. Imitation learning addresses the challenge of designing task-specific reward functions by learning a policy from examples by explicitly tracking the sequence of target poses in the motion clip [Liu and Hodgins 2018; Peng et al. 2018]. While this technique can imitate a single motion clip, it becomes difficult to scale without including high-level motion planners [Bergamin et al. 2019; Lee et al. 2021b,a; Park et al. 2019; Won et al. 2020, 2021; Zhang et al. 2023] or pose-based control using model-based RL [Fussell et al. 2021; Yao et al. 2022] or behavioural cloning [Won et al. 2022], which is prone to drift if small amounts of demonstrations are available [Ross et al. 2011]. Recently, methods based on generative adversarial imitation learning (GAIL) have shown to be an appealing alternative [Bae et al. 2023; Hassan et al. 2023; Lee et al. 2022; Peng et al. 2022, 2021; Xu and Karamouzas 2021], where an adversarial discriminator is trained to serve as an objective function for training a control policy to imitate the demonstrations. We make use of this technique to learn highly diverse and athletic movements. However, while these methods are able to generate diverse and natural looking movements, they lack bio-mechanical insights which are important for movement realism.

Muscoskeletal Methods and Biomechanical Cumulative Fatigue. Several works [Ackermann and van den Bogert 2012; Anderson and Pandy 2001; Geyer and Herr 2010; Ijspeert et al. 2007; Maufroy et al. 2008; Taga 1995; Thelen et al. 2003] developed musculoskeletal models that use biomimetic muscles and tendons to simulate a variety of human and animal motions. Controlling a muscle-based virtual characters was also explored in computer animation, from upper- [Lee et al. 2018, 2009; Lee and Terzopoulos 2006; Sueda et al. 2008; Tsang et al. 2005], to lower- [Park et al. 2022; Wang et al. 2012], and full-body movements [Geijtenbeek et al. 2013; Jiang et al. 2019; Lee et al. 2019, 2014; Wang et al. 2012]. Such methods are computationally expensive, especially for interactive applications such as games. Jiang et al. [2019] convert an optimal control problem in the muscle actuation space to an equal problem in the joint-actuation space. The generated torque limits do not take accumulated fatigue variation over time into account. Muscoskeletal approaches to predicting muscle fatigue are based on detailed muscle activation patterns [Ding et al. 2000; Giat et al. 1996, 1993]. These approaches incorporate fatigue as a modifier of relatively complex muscle models. While they can provide realistic predictions of forces for isolated muscles, they are cumbersome for joint or whole body applications. In contrast, Liu et al. [2002] proposed a computationally efficient motor unit (MU)-based fatigue model, using three muscle activation states to estimate perceived biomechanical fatigue: resting, activated and fatigued. Improving upon this model, Xia and Frey Law [2008] introduced a Three-Compartment Controller (3CC) model for dynamic load conditions, eliminating the need for explicit modeling of muscle actuators.

Cumulative Fatigue Modeling in Simulated Characters. To the best of our knowledge little work has been done in this area. Kider Jr et al. [2011] captured extensive amounts of motion capture and biosignal data, including EKG, BVP, GSR, respiration, and skin temperature, to estimate fatigue of human characters. However, capturing data for all variances is time-consuming and expensive. Komura et al. [2000] make use of a muscoskeletal model, which re-targets existing motion clips to fatigued animations automatically using the musculoskeletal fatigue model Giat et al. [1996, 1993] for lower-body movements. While they achieve variance over time based on biomechanically accurate fatigue assumptions, their method needs an expensive muscoskeletal-model and cannot account for any emergent recovery behaviours. Cheema et al. [2020] use a Three-Compartment-Controller (3CC-*r*) model [Looft et al. 2018; Xia and Frey Law 2008] in a fatigue-related reward function, which does not require expensive modeling and simulation of muscle-tendons. They predict ergonomic differences of user interface configurations with a single arm model in a pointing task. However, they do not consider acrobatic full-body movements. Additionally, none of the mentioned works indicate the emergence of rest behaviours to the same extent as our method and have only been applied to carefully crafted single limb settings with limited movements.

3 PRELIMINARIES - 3CC MODEL

We first review how cumulative fatigue can be modeled efficiently using only joint actuation torques with a 3CC-model [Looft et al. 2018; Xia and Frey Law 2008], which has been used for ergonomic assessment of endurance times and fatigue in biomechanics [Frey-Law et al. 2012b] and HCI [Cheema et al. 2020; Jang et al. 2017].

Motor Units. The 3CC model assumes motor units (MUs) to be in one of three possible states (compartments): 1) *active* – MUs contributing to the task 2) *fatigued* – fatigued MUs without activation 3) *resting* – inactive MUs not required for the task.

These are usually expressed as a percentage of maximum voluntary contraction (*%MVC*), which can practically be expressed as percentage of maximum voluntary force (*%MVF*) or torque (*%MVT*). Rested MUs (M_R) become activated (M_A) once a target load (TL) needs to be held. Active MUs are then directly contributing to the task. Once an MU is activated, its force decays over time and becomes fatigued (M_F). An initial non-fatigued state starts out with $M_{R_0} = 100\%$ and $M_{F_0} = M_{A_0} = 0\%$. The following system of equations describes the change of rate over time t for each compartment:

$$\frac{\partial M_A}{\partial t} = C(t) - F \cdot M_A \tag{1a}$$

$$\frac{\partial M_R}{\partial t} = -C(t) + R_r \cdot M_F \tag{1b}$$

$$\frac{\partial M_F}{\partial t} = F \cdot M_A - R_r \cdot M_F \tag{1c}$$

Here, R_r is defined as

$$R_r = \begin{cases} r \cdot R & \text{if } M_A \geq TL \\ R & \text{else,} \end{cases} \tag{2}$$

where F and R denote the fatigue and recovery coefficients, and r as an additional rest recovery multiplier for intermittent tasks [Looft et al. 2018]. A change in F denotes the change of rate in fatigue, whereas a change in R indicates the overall rate of recovery, as well as an upper bound for maximal fatigue and loads that can be held indefinitely. For example, $R = F \cdot 0.2$ indicates that 20% MVC can be held indefinitely, which would be in accordance to empirical studies [Rohmert 1960]. In this case the limit of M_F would be at 80%. An increase of r indicates an increased recovery rate during tasks with intermittent rest periods when $r > 1$. $C(t)$ in Eq. (1a) and (1b) is a bounded proportional controller, which produces the force required for the target load (TL) by controlling the size of M_A and M_R.

Motor Activation-Deactivation Drive C(t). To obtain behaviours matching muscle physiology – e.g., active MUs decaying over time – control theory is applied. Therefore, $C(t)$ is introduced as a muscle activation-deactivation drive between rested and active MUs:

$$C(t) = \begin{cases} L_R \cdot (TL - M_A) & \text{if } M_A \geq TL \\ L_D \cdot (TL - M_A) & \text{if } M_A < TL \text{ and } M_R > TL - M_A \\ L_D \cdot M_R & \text{if } M_A < TL \text{ and } M_R \leq TL - M_A. \end{cases} \quad (3)$$

The three cases can be described in the following way:

- *Case 1:* If there are more active motor units M_A than required for the target load TL, then M_A decays and M_R increases in Eq. (1), which makes the muscle go into a recovery state. In this case $C(t)$ becomes negative.
- *Case 2:* When there are not enough active motor units M_A compared to the required target TL but the difference is smaller than the available rested MUs M_R, then rested MUs M_R become active MUs M_A. In this case $C(t)$ is positive and greater or equal than $F \cdot M_A$ in Eq. (1a).
- *Case 3:* When there are not enough active motor units M_A compared to the required target TL and not enough rested MUs M_R to compensate the discrepancy, the muscle starts to fatigue and the target load cannot be held any longer. In this case, $C(t)$ is positive but smaller than $F \cdot M_A$ in Eq. (1a). Here, M_A decays and M_R becomes (near) zero.

L_D and L_R are muscle force development and relaxation factors, which describe the sensitivity towards the target load. Since the time course of either is negligible compared to the time course of fatigue (e.g. varying L_D/L_R from 2 to 50, or a change of 2500%, only alters the endurance time by 10%), Xia and Frey Law [2008] set the same arbitrary value to each $L_D = L_R = 10$.

Residual Capacity. Residual Capacity (RC) describes the remaining motor strength capabilities or stamina due to fatigue in percent. 0% indicates no strength reserve, and 100% indicates full non-fatigued strength:

$$RC(t) = M_A + M_R = 100\% - M_F. \quad (4)$$

While the 3CC model is a great analysis tool for fatigue and endurance time estimation [Cheema et al. 2020; Frey Law and Avin 2010; Jang et al. 2017], it does not directly lend itself to create full-body character animations modeling the described fatigue effects in Sec. 1. To approach this challenge, we leverage Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon 2016; Peng et al. 2022, 2021; Torabi et al. 2018] and constrain the action space based on the computed Residual Capacity of the 3CC model, which we describe in more detail in the following section.

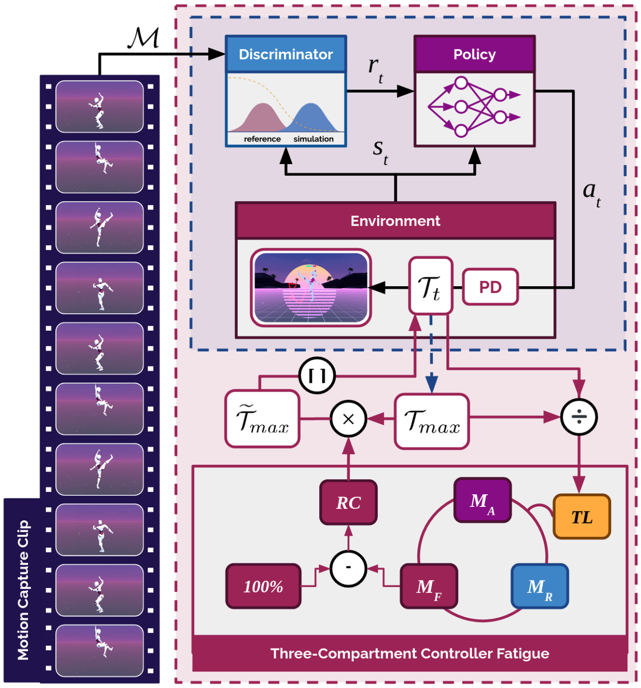

Figure 2: Overview. Our framework consists of two stages: A pre-training stage where the policy learns to imitate the motion clip.Then a transfer learning stage, where an adaptive fatigue policy is learned by constraining the output torques of the PD-controllers based on the computed torque bounds by the Residual Capacity (RC) from the 3CC-model.

4 FATIGUE MODELING FOR CHARACTER ANIMATION

Our method takes as input a motion clip $\mathcal{M}$ of the full-body skeleton of the humanoid character represented by a sequence of poses $\hat{q}_t$. This motion clip does *not* contain varying fatigue levels. Nonetheless, our goal is to generate an animation that mimics the original behaviour while the character fatigues over time and learns plausible recovery strategies. At the technical core, we explore a two-step approach that effectively blends ideas from biomechanical cumulative fatigue modeling, biologically-inspired torque limit constraining, and Deep Reinforcement Learning to enable the emergence of realistic symptoms of fatigue in character animation (see Fig. 2).

We first pre-train the policies on the reference motions to estimate the maximum constant torque bounds $\mathcal{T}_{max}$ across the tasks. The actions $\mathbf{a}_t := (\mathbf{u}_t, \beta_t)$ at time t from the policy π specify target positions $\mathbf{u}_t$ for PD-controllers positioned at each of the character's joints and a stiffness and damping multiplier β_t, similar to [Yuan et al. 2021], which we query at the policy frequency. Modulating stiffness and damping introduces the possibility for the character to relax and tense its whole body, which is appropriate for the context of fatigue modeling, as fatigued virtual characters using proportion-derivative (PD) controllers with fixed stiffness/damping parameters may choose overly stiff/conservative motions instead of relaxed motions. The output torques are then applied to the character physics simulation (Sec. 4.1). Once an expert policy has been learned, we use transfer learning to learn a policy, which is able to adapt to fatigue by constraining the torque-bounds over time based on RC computed by the 3CC fatigue modules resulting in $\widetilde{\mathcal{T}}_{max}$ (Sec. 4.1). This forces the policy to handle lower torque levels and discover fatigue and rest behaviours in attempt to fulfill the task based on the given constraints. Importantly, the model can be trained on a single (F, R, r)-triplet and adapt to novel triplets during inference, which makes training more efficient. Once trained, the agents exhibit unseen fatigued movement patterns and unseen rest recovery strategies emerge to overcome the loss of strength.

4.1 Imitation Objective and Torque-Estimation

In this section, we describe the pre-training. We start with a general formulation of a DRL problem, then continue with the imitation objective, and finally describe the constant torque-bound estimation.

RL Problem Formulation. At each time step t, the agent observes a state $\mathbf{s}_t$ based on its environment observations and samples an action $\mathbf{a}_t$ from a policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$ in accordance to the observed state, which leads to a new state $\mathbf{s}_{t+1}$ and a reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$. The agent’s objective is to maximize its expected discounted return $J(\pi)$ [Sutton et al. 1998]

$$J(\pi) = \mathbb{E}_{p(\tau|\pi)}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right], \tag{5}$$

where $p(\tau|\pi) = p(\mathbf{s}_0)\prod_{t=0}^{T-1} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t|\mathbf{s}_t)$ represents the likelihood of the trajectory $\tau = \left\{(s_t, a_t, r_t)_{t=0}^{T-1}, s_T\right\}$ under a policy π. Here, $p(\mathbf{s}_0)$ denotes the initial state distribution, T is the time horizon of a trajectory, and $\gamma \in [0, 1]$ is the discount factor. To design a reward objective, which can imitate diverse athletic movements, we leverage Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon 2016; Peng et al. 2022, 2021].

Imitation Objective. In GAIL, the objective to imitate a given task is modeled as a discriminator $D(\mathbf{s}, \mathbf{a})$, which is trained to predict whether a given state $\mathbf{s}$ and action $\mathbf{a}$ is sampled from the demonstrations $\mathcal{M}$ or generated from the policy π [Ho and Ermon 2016]. This formulation of GAIL requires access to the demonstrator’s actions, which, however, are not given when only motion clips are provided as demonstrations. Similar to Torabi et al. [2018], we train the discriminator on state transitions $D(\mathbf{s}, \mathbf{s}')$ instead of state-action pairs $D(\mathbf{s}, \mathbf{a})$ to overcome this limitation:

$$\underset{D}{\operatorname{argmin}} -\mathbb{E}_{p^{\mathcal{M}(\mathbf{s},\mathbf{s}')}}\left[\log\left(D(\mathbf{s}, \mathbf{s}')\right)\right] - \mathbb{E}_{p^{\pi(\mathbf{s},\mathbf{s}')}}\left[\log\left(1 - D(\mathbf{s}, \mathbf{s}')\right)\right], \tag{6}$$

where $p^{\mathcal{M}(s,s')}$ and $p^{\pi(s,s')}$ denote the likelihoods of observing a state transition from state $\mathbf{s}$ to $\mathbf{s}'$ in the dataset $\mathcal{M}$, and following the policy π, respectively. Additionally, we incorporate the gradient penalty regularizers [Peng et al. 2021]. The discriminator is then trained using the following objective

$$\underset{D}{\operatorname{argmin}} - \mathbb{E}_{p^{\mathcal{M}(\mathbf{s},\mathbf{s}')}}\left[\log\left(D(\mathbf{s}, \mathbf{s}')\right)\right] - \mathbb{E}_{p^{\pi(\mathbf{s},\mathbf{s}')}}\left[\log\left(1 - D(\mathbf{s}, \mathbf{s}')\right)\right] + w_{\mathrm{gp}}\mathbb{E}_{d^{\mathcal{M}(\mathbf{s},\mathbf{s}')}}\left[\|\nabla_\phi D(\phi)|_{\phi=(\mathbf{s},\mathbf{s}')}\|^2\right], \tag{7}$$

where w_{gp} is a gradient penalty coefficient. Akin to Peng et al. [2022], we use the imitation objective for an adversarial imitation policy by defining the reward objective in Eq. (5) as

$$r_t = -\log(1 - D(\Phi(\mathbf{s}_t), \Phi(\mathbf{s}_{t+1}))), \tag{8}$$

where $\Phi(\mathbf{s}_t)$ denotes a feature map based on the state space $\mathbf{s}_t$.

Torque Estimation. We use the following PD-controller formulation to estimate the joint torque bounds and actuation torques [Featherstone 2014; Yuan et al. 2021] at every control query:

$$\mathcal{T}_d := \beta_d k_{\mathrm{p}}^d(u_d - \hat{u}_d) - \beta_d k_{\mathrm{d}}^d \dot{\hat{u}}_d \tag{9}$$

where u_d denotes the target orientation and β_d a stiffness and damping modulation parameter given from the policy’s action $a_d := (u_d, \beta_d)$. $\hat{u}_d$ denotes the current orientation of the DoF d and $\dot{\hat{u}}_t$ the current velocity. k_{p} and k_{d} specify constant stiffness and damping parameters. The pre-training has the nice side-effect that we can automatically estimate maximum torques as

$$\mathcal{T}_{max_d} := \max(\mathcal{T}_d, \mathcal{T}_{max_d}) \tag{10}$$

with $\mathcal{T}_{max_{d_0}} = 0$, which avoids manual or grid search for this hyperparameter. Additionally, we consider physiological symmetries by ensuring that two symmetric joints, e.g. left and right elbow, have the same value by computing the *minimum* between the two, as lower energy movements tend to look more natural [Yu et al. 2018]. In addition, we found we are able to greatly reduce outliers for potential $\mathcal{T}_{max_d}$ candidates. We note that our method works for hand-crafted torque limits, as well as limits which depend on a single and multiple motions. In these cases, just the rate of fatigue and recovery for an (F, R, r) setting changes.

Transfer Learning: Fatigue-based Torque Limits. Inspired by Jiang et al. [2019], we limit constant torque-bounds to nonlinear state-dependent limits in the joint actuation space. While their method allows for bio-mechanically enhanced torque-based actuation, it does not allow for variance over time from cumulative fatigue. Thus, we multiply the residual capacity for each 3CC-model with the maximum torque bounds found in the previous stage and then use transfer learning to make the policy adapt to the loss of strength as explained next.

Fatigued Torque Bounds. We assume the target load TL to be the incoming joint actuator torques of the respective PD-controller of each DoF, which the character requires to reach the target position. Each DoF d is modeled by a 3CC model as

$$TL_d(\mathcal{T}_d) = \frac{\mathcal{T}_d}{\mathcal{T}_{max_d}} \cdot 100\% \tag{11}$$

being the ratio of the incoming actuator torque $\mathcal{T}_d$ computed by the respective PD-controller and the constant maximum torque-bound found in the previous step representing the percentage of maximum voluntary contraction $\%MVC$. M_{A_d}, M_{R_d} and M_{F_d} are computed in accordance to the incoming target load per DoF using Eq. (1)–(3). To estimate the fatigued torque bounds $\widetilde{\mathcal{T}}_{max_d}$, we leverage the residual capacity RC as a time-varying multiplier to the previously found torque bound limits:

$$\widetilde{\mathcal{T}}_{max_d} = RC_d \cdot \mathcal{T}_{max_d}, \tag{12}$$

where $RC_d = 100\% - M_{F_d}$ (see Sec. 3). The final fatigued torque $\widetilde{\mathcal{T}}_d$ applied to the environment for each DoF is computed by clipping the incoming joint actuator torque $\mathcal{T}_d$ within $RC_d \cdot \left[-\mathcal{T}_{max_d}, \mathcal{T}_{max_d}\right]$ with $\widetilde{\mathcal{T}}_d$ being defined as

$$\widetilde{\mathcal{T}}_d \in \left[-\widetilde{\mathcal{T}}_{max_d}, \widetilde{\mathcal{T}}_{max_d}\right] = \left[-RC_d \cdot \mathcal{T}_{max_d}, RC_d \cdot \mathcal{T}_{max_d}\right]. \tag{13}$$

Transfer Learning. Simply reducing the joint actuator torques of an agent with a policy trained with full torque-bounds will let the agent fall as the policy never learned to deal with loss of strength. Thus, we apply transfer learning, where the policy is trained on the time-varying torque outputs based on the Residual Capacity. While R and F values are joint specific [Frey-Law et al. 2012b; Looft et al. 2018] accounting for varying endurance times [Frey Law and Avin 2010], we use one F, R and r for the whole character as a design choice for usability and simple interactivity. We note that despite training on one (F, R, r) triplet our policy is able to adapt to new pairs during inference. This is due to the fact that a single (F, R, r)-triplet alone already corresponds to multiple levels of Residual Capacity over time – making it easy for the agent to adapt to another combination even though the change of rate of RC may differ. As such our method can be viewed as a framework for generating a range of motor skills from a single motion clip [Lee et al. 2021a], where the skills are parameterized by fatigue.

5 MODEL REPRESENTATION

States and Actions. We evaluate our framework using the 28 DoF humanoid character from Peng et al. [2021], as well as their state-space $\mathbf{s}_t$ representation in the IsaacGym implementation with the inclusion of the character's local root rotation [Peng et al. 2022], as well as the fatigued motor units M_F in the observation space, totalling a state space of 133 dimensions. The character's local coordinate frame is defined as in [Peng et al. 2022, 2021]. Each action $\mathbf{a}_t := (\mathbf{u}_t, \beta_t)$ specifies the target rotations $\mathbf{u}_t$ and stiffness/damping multiplier β_t) for the PD-controllers at each of the character's joints. This results in a 29D action space – one action per DoF, as well as the stiffness/damping multiplier. M_F is randomized during expert training to make the policy agnostic to it before it adapts to it in the fine-tuning stage.

Network Architecture. The policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$ is represented as a neural network, with the action distribution modeled as a Gaussian, where the state-dependent mean $\mu(\mathbf{s}_t)$ and the diagonal covariance matrix Σ are specified by the network output $\pi(\mathbf{a}_t|\mathbf{s}_t) = \mathcal{N}(\mu(\mathbf{s}_t), \Sigma)$. The mean is specified by an MLP consisting of two fully-connected hidden layers of 1024 and 512 Rectified Linear Units (ReLU) [Nair and Hinton 2010], followed by a linear output layer. The values of the covariance matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots)$ are manually specified [Peng et al. 2021] and fixed over the course of training. The value function $V(\mathbf{s}_t)$ and the discriminator $D(\mathbf{s}_t, \mathbf{s}_{t+1})$ are modeled by separate networks with a similar architecture as the policy.

6 EVALUATION AND RESULTS

We evaluate our method on five diverse movement skills – backflip, cartwheel, hopping and locomotion from the CMU dataset [CMU] provided in the Isaac Gym [Makoviychuk et al. 2021] environment, as well as the 360 tornado kick from the SFU dataset [SFU]. The experiments below evaluate the following aspects of our method: First, compared to constant torque limits, our state and time-varying torque limits push the policy network towards movement strategies and patterns not present in the input data $\mathcal{M}$ to overcome the loss of strength. Second, the found recovery strategies resemble human-like strategies for resting. All experiments are carried out using the high-performance GPU-based physics simulator Isaac Gym [Makoviychuk et al. 2021]. During training 4096 environments are simulated in parallel on a single NVIDIA V100 GPU with a simulation frequency of 120Hz, while the policy operates at 30 Hz. All neural networks are trained using PyTorch [Paszke et al. 2019]. Gradient updates are performed via Proximal Policy Optimization [Schulman et al. 2017] with a learning rate of 5×10^{-5}. We use an episode length of 300 during pre-training and an episode length of 1000 during fatigue transfer to learn the accumulation of fatigue. The gradient penalty coefficient w_{gp} in Eq. (6) is set to 0.2 for all but locomotion, for which it is set to 5 [Makoviychuk et al. 2021]. Additional hyper-parameter settings and implementation details can be found in the supplementary document. We use the "Humanoid AMP" [Peng et al. 2021] character provided in Isaac Gym with 28 internal DoFs and its corresponding rigid body and joint properties. Stiffness and damping parameters were set to custom values in accordance to the realistic proportions of a real-life human male. M_A, M_F and M_R are randomized at every environment reset.

6.1 Fatigue Training

We first note that simply taking a pre-trained policy and adjusting the torque limits during inference only lets the character fall into a termination state and not learn any realistic recovery strategies because the character never learned to deal with less torque as can be seen in Fig. 3. Thus, we employ a simple transfer learning procedure where we train the expert policies for several iterations until stable behaviours arise. We disable fatigue behaviour during this pre-training phase. As the observation still contains M_F values for the expert policy training, we ensure that the expert policies are agnostic to the M_F values by randomizing the M_F values at every environment step during this phase, as a strategy for domain randomization [Tobin et al. 2017]. More specifically, we train these expert policies for 2000 iterations for running and 4000 for others. For transfer learning, we apply 2000 iterations of additional training for each motion, using the corresponding expert policy as the warm-starting point. We input $F = 1, R = 0.01, r = 1$ for all transfer learning iterations. The reference torque estimates for computing TL values are given by the expert policies. During the transfer learning phase, we randomize the fatigue state at uniform at each reset of the training episode, as to capture as much variability of the fatigue state as possible while observing the input (F, R, r)-triplet. We found that we can train on a single (F, R, r)-triplet but test on a variety of combinations (Fig. 1 and 11) if fast enough gradual loss of strength due to fatigue can be observed during training time, as well as some form of increase in strength during recovery periods.

6.2 Fatigue Movements and Recovery Strategies

The loss of strength modeled by our method leads to new movement patterns for the rest recovery strategies, as well as the divergence from the input motion the model was trained on. We found the following behaviours, which compensate for the loss of strength: **Waiting** by standing or doing a couple of steps as observed in the cartwheel (Fig. 5), tornado kick (Fig. 1) and backflip (Fig. 11). Fig. 1, 4, 5 and 11 show how the agent regains strength during such rest periods; a **change of performance**, e.g. decrease of height of jumps, especially observable in the hopping and tornado kick motions (Fig. 12); a **change of motion style**, e.g. with increased tucking and knee-bending behaviours in dynamic motions, such as backflip (Fig. 10); and **compensation of forces** with movements requiring a lot of momentum such as the cartwheel or backflip by trembling or requiring more suspension (Fig. 9). Additionally, a **reduction of number of repetitions** (Fig. 13) **and speed** (Fig. 14) can be observed. Fig. 4 shows a comparison between the original 3CC model resulting from the cartwheel motion (*left*), as well as our modification for animation (*right*). Note how the fatigue M_F decreases during the rest period between 6 and 19 seconds (*left*), and increases with each cartwheel. A cartwheel is indicated by the three spikes in TL and M_A in the beginning, as well as the spike at 20 seconds – in correspondence to the 4 cartwheels in Fig. 5. For animation, we make use of the residual capacity $RC = 100\% - M_F$ as a strength multiplier for the constant torque bounds. The applied torques (dashed blue line) become similar to M_A when being cut off during fatigue and stay equal to TL during non-fatiguing periods.

360 Tornado Kick. During fatigue the character learns to kick with lower foot height and a reduced distance between the legs, which can be best observed in Fig. 1 and Fig. 12. As the motion requires a lot of strength and flexibility in the leg region, the agent learns to recover by lowering the jump, as well as waiting between jumps. The most affected joints by fatigue are the kicking knee, as well as the arm joints required to gain the momentum for the jump (Fig. 7).

Backflip. The backflips become lower when fatigued with increase in tucking (Fig. 10) and compensation of forces (Fig. 9 *bottom*). After rest, the agent is able to do backflips with better forms again but cannot do as many anymore before it needs to rest again (Fig. 11).

Cartwheel. We find that the character learns to stand or walk in order to rest from the fatigue after a repetition of cartwheels (Fig. 5). Furthermore, with each repetition, before the character is able to rest fully, the leg height becomes lower.

Hopping. The most apparent changes in the hopping motion during fatigue are the jump height/length and frequency (Fig. 12 *top*).

Locomotion. The most apparent change over time is the reduction of speed as well as the change of stride length (Fig. 14).

6.3 Learning Diverse Fitness Levels in One Policy

With the fatigue model driving the behaviour of the policy via the fluctuation of M_F, the policy is capable of handling a variety of fatigue states it encounters during deployment. Here, we highlight a key advantage of using the 3CC fatigue model by demonstrating the capability to model different fitness levels using the same character and simply manipulating the parameters of the 3CC model at deployment. More specifically, the policy outputs a full spectrum from a high-stamina to a low-stamina character behaviours as a response to intuitive parameter adjustments at runtime. Figures 1, 8 and 11 juxtapose three different scenarios rendered from deploying the same policy in the same initial state: a non-fatigable character for qualitative baseline (top), a high-stamina character (middle), and a low-stamina character. We further emphasize how varying (F, R, r) parameters over time can be used for the 3CC for more fine-grained control. Simply put, we achieve the capability to capture the diversity of fitness levels with the same reference motion, character specification, and policy, which is similar to a *mixture of experts* policy where experts corresponding to different fitness levels emerge depending on the (F, R, r) input without any need for fatigued reference motions. Additionally, we show that our method can also be used to analyze which joints are being most affected and to what degree by fatigue (Fig. 7).

Table 1: Comparison between our method and AMP. Distance comparison against motion capture data between our trained model and Peng et al. [2021]. The distance describes the average Euclidean distance [and variance] of the normalized joint angles between a generated frame and the closest frame in motion file $\mathcal{M}$. The distance is averaged over an episode.

	Backflip	Cartwh.	Hopping	Locomot.	360 Kick
AMP	0.18[0.06]	0.24[0.07]	0.16[0.03]	0.23[0.05]	0.24[0.08]
Ours	0.44[0.36]	0.25[0.09]	0.32[0.33]	0.36[0.28]	0.32[0.17]

6.4 Ablations and Comparisons

We ablate our method with and without the torque coefficient β (Fig. 6) showing that it improves upon motion smoothness when limiting torques. Not observing fatigue leads additionally to low fidelity motions as can be seen in the supplementary video. To validate our method we show that our model is able to switch from a running clip to a walking clip without motion blending solely based on the fatigue (supp. video). We further compare our method against two baselines: 1) Against a GAIL-baseline based on AMP [Peng et al. 2021]; 2) Against the reward-based fatigue model by Cheema et al. [2020]. For the former we use the implementation in Isaac Gym [Makoviychuk et al. 2021], whereas for the latter we fine-tune our pre-trained model with their fatigue-based reward without any torque limitation.

GAIL without Fatigue Control. We observe in Tab. 1 that our model is able to synthesize novel fatigued behaviours and recovery strategies not present in dataset, whereas Peng et al. [2022, 2021] are solely able to imitate existing motion capture disregarding any variance over time due to cumulative fatigue (see also supplemental video). In contrast to that our method allows for interactive control of fatigue and varying degrees of fitness levels over time by changing (F, R, r) values during inference (see Fig. 1). Previous methods [Peng et al. 2022, 2021] are not able to provide such interactive control and varying degrees of fitness levels in one policy.

Fatigue-based Reward. The closest method [Cheema et al. 2020] to our's uses a reward based on the difference between M_A and

TL. To compare our torque-limit-based method to their reward-based method, we use their reward for fatigue fine-tuning of our GAIL-baseline. The reward is added to r_t in Eq. (8). The results can be observed in Fig. 15. We show that despite their method being based on the 3CC model and resulting in accurate fatigue analysis, it is not able to synthesize correct fatigued behaviours over time, especially when combined with a GAIL-based reward. Fig. 15 *(forth subplot from the top)* shows that while *RC* is high, the agent does a lot of fatigued kicks with lower heights. These however become increasingly higher as the *RC* lowers, since then the agent obtains most of its reward from the imitation reward. The agent furthermore then seems to exploit the imitation learning policy by doing higher jumps than that are in the dataset since the agent can “rest" during the fall of the increased air time. Using the opposite of this reward instead leads to imitation learning in the beginning and faster motions later on, which require even more energy. This effect holds true for several hyperparameter combinations for their reward. While Cheema et al. [2020] have used their method to merely enhance the naturalness of pointing movements during rest periods, as well as to analyse and predict ergonomic differences of user interface configurations, we found that combining their method with an imitation learning policy fails to actually synthesize expected fatigue behaviour and requires careful tuning of reward parameters. Additionally, we noticed that a reward-based method does not give an intuitive control over the 3CC-parameters as it is the case for our torque-limit based model which is described in Sec. 6.1. Fig. 15 also shows that a method solely based on reward does not correspond to the actual strength capabilities that should be left with a given (F, R, r) setting, as it is not a hard constraint. Setting $R = 0.2F$ means that at maximum fatigue 20% strength should be left. However, a reward-based method does not account for such physiological restrictions, while our joint actuation torques actually correspond to the level of residual capacity.

7 CONCLUSION

We presented the first full-body approach for physics-based 3D humanoid motion synthesis with fatigue. Our experiments demonstrate the emergence of realistic fatigued movements and recovery behaviours for interactive athletic animations that are difficult to produce with previous techniques (especially without muscoskeletal simulation). Our torque-based fatigue simulation system can be efficiently used for real-time interactive 3D virtual character animation, opening up new possibilities for games and animation tools. Additionally, we show in the supplementary material that due to the morphology-agnostic nature of the 3CC-model our method can be applied to characters of any physiology, as long as a measure of $\%MVC$ as a function of either torque or force can be provided. It can further be easily extended with additional task-rewards such as target goals or orientations. We believe this work opens many exciting pathways for biomechanical simulations and physics-based character animation, particularly in terms of discovering automatic emergent behaviours for intelligent agents.

ACKNOWLEDGMENTS

Noshaba Cheema was supported from the EIC Pathfinder grant Carousel+ (101017779) and an IMPRS fellowship. Rui Xu was supported from the BMWi grant KAI (19A21022C). Nam Hee Kim received support from the FCAI Open Application grant and an NSERC Postgraduate Scholarship (567794-2022). The authors from MPII were supported by the ERC Consolidator Grant 4DRepLy (770784). We thank Erik Herrmann and Han Du for their retargeting framework and for providing invaluable feedback, alongside Jaakko Lehtinen. We also wish to express our appreciation to Ahmed 'The Music Jnn' Hadrich for the sound design; as well as to Laura Morales, Seth Izen and Tobias Cheema for their additional support with the supplementary video and figures.

REFERENCES

Marko Ackermann and Antonie J van den Bogert. 2012. Predictive simulation of gait at low gravity reveals skipping as the preferred locomotion strategy. *Journal of biomechanics* 45, 7 (2012), 1293–1298.

Frank C Anderson and Marcus G Pandy. 2001. Dynamic optimization of human walking. *J. Biomech. Eng.* 123, 5 (2001), 381–390.

Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. 2023. PMP: Learning to Physically Interact with Environments Using Part-Wise Motion Priors. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 64, 10 pages. https://doi.org/10.1145/3588432.3591487

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2019. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528* (2019).

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).

Gunnar AV Borg. 1982. Psychophysical bases of perceived exertion. *Medicine & science in sports & exercise* (1982).

Michael Büttner and Simon Clavet. 2015. Motion matching-the road to next gen animation. *Proc. of Nucl. ai* 2015, 1 (2015).

Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*.

CMU. 2003. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/

Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. 2010. Generalized biped walking control. *ACM Transactions On Graphics (TOG)* 29, 4 (2010).

H Onan Demirel and Vincent G Duffy. 2007. Applications of digital human modeling in industry. In *International Conference on Digital Human Modeling*. Springer, 824–832.

Jun Ding, Anthony S Wexler, and Stuart A Binder-Macleod. 2000. A predictive model of fatigue in human skeletal muscles. *Journal of applied physiology* 89, 4 (2000), 1322–1332.

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

Martin L Felis and Katja Mombaur. 2016. Synthesis of full-body 3-d human gait using optimal control methods. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1560–1566.

Mihai Fieraru, Mihai Zanfir, Silviu-Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. 2021. AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Laura A Frey Law and Keith G Avin. 2010. Endurance time is joint-specific: a modelling and meta-analysis investigation. *Ergonomics* 53, 1 (2010), 109–129.

Laura A Frey-Law, Andrea Laake, Keith G Avin, Jesse Heitsman, Tim Marler, and Karim Abdel-Malek. 2012a. Knee and elbow 3D strength surfaces: peak torque-angle-velocity relationships. *Journal of applied biomechanics* 28, 6 (2012), 726–737.

Laura A Frey-Law, John M Looft, and Jesse Heitsman. 2012b. A three-compartment muscle fatigue model accurately predicts joint-specific maximum endurance times for sustained isometric tasks. *Journal of biomechanics* 45, 10 (2012), 1803–1808.

Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. Supertrack: Motion tracking for physically simulated characters using supervised learning. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.

Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)* 32, 6 (2013).

Hartmut Geyer and Hugh Herr. 2010. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering* 18, 3 (2010), 263–273.

Yohanan Giat, Joseph Mizrahi, and Mark Levy. 1996. A model of fatigue and recovery in paraplegic's quadriceps muscle subjected to intermittent FES. (1996).

Yohanan Giat, Joseph Mizrahi, Mark Levy, et al. 1993. A musculotendon model of the fatigue profiles of paralyzed quadriceps muscle under FES. *IEEE Transactions on Biomedical Engineering* 40, 7 (1993), 664–674.

Perttu Hämäläinen, Joose Rajamäki, and C Karen Liu. 2015. Online control of simulated humanoids using particle belief propagation. *ACM Transactions on Graphics (TOG)* 34, 4 (2015).

Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. 2023. Synthesizing Physical Character-Scene Interactions. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 63, 9 pages. https://doi.org/10.1145/3588432.3591525

Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1063–1072.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017).

Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. 2007. From swimming to walking with a salamander robot driven by a spinal cord model. *science* 315, 5817 (2007), 1416–1420.

Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3328–3339.

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).

Joseph T Kider Jr, Kaitlin Pollock, and Alla Safonova. 2011. A data-driven appearance model for human fatigue. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 119–128.

Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L Kunii. 2000. Creating and retargetting motion by the musculoskeletal human body model. *The visual computer* 16, 5 (2000), 254–270.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion Graphs. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (San Antonio, Texas) *(SIGGRAPH '02)*. ACM, New York, NY, USA, 473–482. https://doi.org/10.1145/566570.566605

Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. 2002. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 491–500.

Kyungho Lee, Sehee Min, Sunmin Lee, and Jehee Lee. 2021b. Learning time-critical responses for interactive character control. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

Seunghwan Lee, Phil Sik Chang, and Jehee Lee. 2022. Deep Compliant Control. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 23, 9 pages. https://doi.org/10.1145/3528233.3530719

Seyoung Lee, Sunmin Lee, Yongwoo Lee, and Jehee Lee. 2021a. Learning a family of motor skills from a single motion clip. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

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).

Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. 2018. Dexterous manipulation and control with volumetric muscles. *ACM Transactions on Graphics (TOG)* 37, 4 (2018).

Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics (TOG)* 28, 4 (2009).

Sung-Hee Lee and Demetri Terzopoulos. 2006. Heads up! Biomechanical modeling and neuromuscular control of the neck. In *ACM SIGGRAPH 2006 Papers*. 1188–1198.

Yoonsang Lee, Sungeun Kim, and Jehee Lee. 2010. Data-driven biped control. In *ACM SIGGRAPH 2010 papers*.

Yoonsang Lee, Moon Seok Park, Taesoo Kwon, and Jehee Lee. 2014. Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics (TOG)* 33, 6 (2014).

Sergey Levine, Jack M Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. 2012. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (TOG)* 31, 4 (2012).

Jing Z Liu, Robert W Brown, and Guang H Yue. 2002. A dynamical model of muscle activation, fatigue, and recovery. *Biophysical journal* 82, 5 (2002), 2344–2359.

Libin Liu and Jessica Hodgins. 2017. Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)* 36, 3 (2017).

Libin Liu and Jessica Hodgins. 2018. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 4 (2018).

John M Looft, Nicole Herkert, and Laura Frey-Law. 2018. Modification of a three-compartment muscle fatigue model to predict peak torque decline during intermittent tasks. *Journal of biomechanics* 77 (2018), 16–25.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470* (2021).

Christophe Maufroy, Hiroshi Kimura, and Kunikatsu Takase. 2008. Towards a general neural controller for quadrupedal locomotion. *Neural Networks* 21, 4 (2008), 667–681.

Charu M Maurya, Sougata Karmakar, and Amarendra Kumar Das. 2019. Digital human modeling (DHM) for improving work environment for specially-abled and elderly. *SN Applied Sciences* 1, 11 (2019).

Jianyuan Min and Jinxiang Chai. 2012. Motion graphs++ a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG)* 31, 6 (2012).

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*.

Jungnam Park, Sehee Min, Phil Sik Chang, Jaedong Lee, Moon Seok Park, and Jehee Lee. 2022. Generative GaitNet. In *ACM SIGGRAPH 2022 Conference Proceedings*.

Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)* 38, 6 (2019).

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. *Advances in neural information processing systems* 32 (2019).

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR, 2778–2787.

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

Xue Bin Peng, Glen Berseth, and Michiel Van de Panne. 2016. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 35, 4 (2016).

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.* 41, 4, Article 94 (July 2022).

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *arXiv preprint arXiv:2104.02180* (2021).

Marc H Raibert and Jessica K Hodgins. 1991. Animation of dynamic legged locomotion. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 349–358.

Walter Rohmert. 1960. Ermittlung von Erholungspausen für statische Arbeit des Menschen. *Internationale Zeitschrift für angewandte Physiologie einschließlich Arbeitsphysiologie* 18, 2 (1960), 123–164.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 627–635.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

SFU. 2019. SFU Motion Capture Database. https://mocap.cs.sfu.ca/

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)* 39, 4 (2020).

Shinjiro Sueda, Andrew Kaufman, and Dinesh K Pai. 2008. Musculotendon simulation for hand animation. In *ACM SIGGRAPH 2008 papers*.

Richard S Sutton, Andrew G Barto, et al. 1998. Introduction to reinforcement learning. (1998).

Gentaro Taga. 1995. A model of the neuro-musculo-skeletal system for human locomotion. *Biological cybernetics* 73, 2 (1995), 97–111.

Yuval Tassa, Tom Erez, and Emanuel Todorov. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 4906–4913.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. 2017. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems* 30 (2017).

Darryl G Thelen, Frank C Anderson, and Scott L Delp. 2003. Generating dynamic simulations of movement using computed muscle control. *Journal of biomechanics* 36, 3 (2003), 321–328.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 23–30.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158* (2018).

Winnie Tsang, Karan Singh, and Eugene Fiume. 2005. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 319–328.

Kevin Wampler, Zoran Popović, and Jovan Popović. 2014. Generalizing locomotion style to new animals with inverse optimal regression. *ACM Transactions on Graphics (TOG)* 33, 4 (2014).

Jack M Wang, Samuel R Hamner, Scott L Delp, and Vladlen Koltun. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics (TOG)* 31, 4 (2012).

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)* 39, 4 (2020).

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.

Ting Xia and Laura A. Frey Law. 2008. A theoretical approach for modeling peripheral muscle fatigue and recovery. *Journal of Biomechanics* 41, 14 (2008), 3046–3052. https://doi.org/10.1016/j.jbiomech.2008.07.013

Pei Xu and Ioannis Karamouzas. 2021. A GAN-Like Approach for Physics-Based Imitation Learning and Interactive Character Control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4, 3 (2021).

Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.

Yuting Ye and C Karen Liu. 2010a. Optimal feedback control for character animation using an abstract model. In *ACM SIGGRAPH 2010 papers*.

Yuting Ye and C Karen Liu. 2010b. Synthesis of responsive motion using a dynamic model. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 555–562.

KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)* 26, 3 (2007).

Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and KangKang Yin. 2021. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

Wenhao Yu, Greg Turk, and C. Karen Liu. 2018. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics* 37, 4 (2018). https://doi.org/10.1145/3197517.3201397

Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. 2021. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7159–7169.

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. 2023. Simulation and Retargeting of Complex Multi-Character Interactions. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 65, 11 pages. https://doi.org/10.1145/3588432.3591491

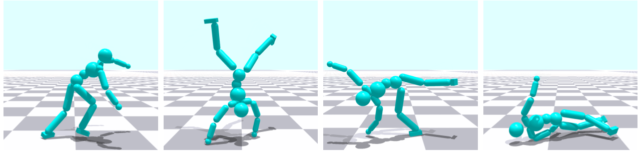

Figure 3: Pre-trained policy with fatigued torque limits during inference without transfer learning results in task failure, since the policy has not learned to deal with using lower torques and forces.

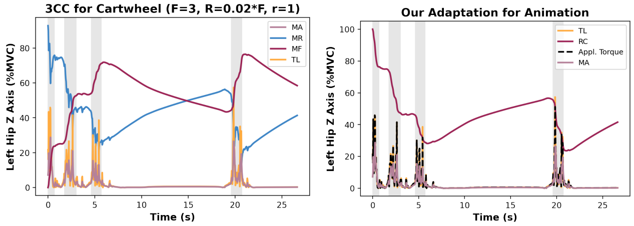

Figure 4: Results of the 3CC model (*left*) and its adaptation (*right*) for the cartwheel. The graphs correspond to the cartwheel motion depicted in Fig. 5. The gray areas indicate a successful cartwheel, whereas the areas between 6s and 19s, as well as after 20s correspond to emerging waiting behaviors to rest from the tiring actions.

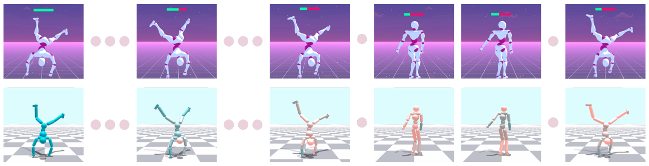

Figure 5: Recovery behaviour of a cartwheel motion. After three cartwheels the agent becomes tired, rests, and does another cartwheel when enough stamina is regained.

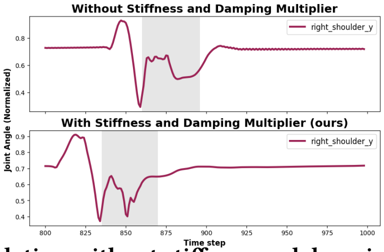

Figure 6: Ablation without stiffness and damping multiplier (*top*) and with *(bottom, ours)* for the shoulder joint during the backflip motion. While the agent is standing and supposed to rest, the shoulder starts to jitter without our torque coefficient β, while with it the motion is smooth and still (*bottom*).

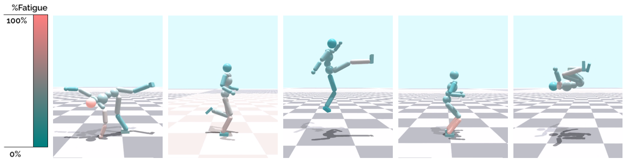

Figure 7: Most affected joints from fatigue. *From left to right:* Cartwheel, Locomotion, 360 Tornado Kick, Hopping, and Backflip.

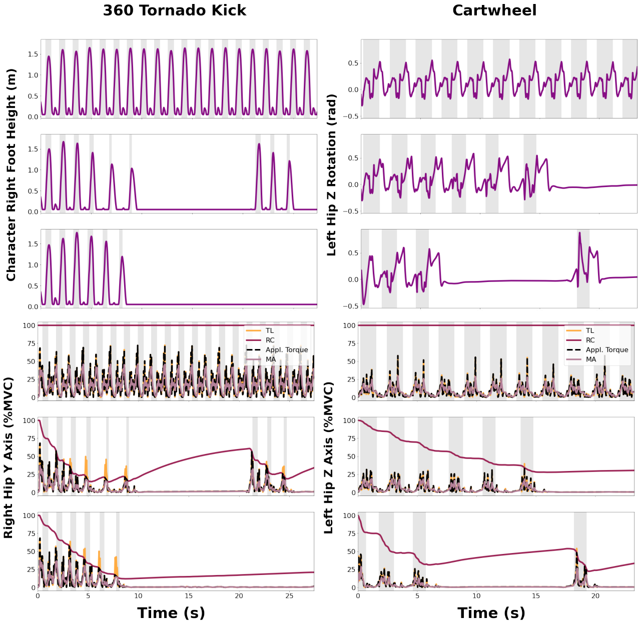

Figure 8: Movement patterns (*top*) and %MVC (*bottom*) for a prominent DoF. The models show the following (F, R, r) settings (from top to bottom): Tornado Kick – (0, 0, 0), (2.0, 0.05, 1.0), (1.0, 0.01, 1.0); Cartwheel – (0, 0, 0), (1.0, 0.01, 1.0), (3.0, 0.02, 1.0) The graphs at the bottom show the results for our modification of the 3CC model for animation.

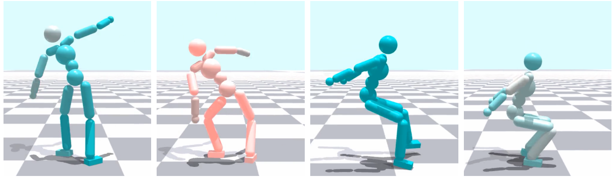

Figure 9: Compensation of forces. The character compensates forces for movements requiring a lot of momentum such as the cartwheel or backflip by trembling or requiring more suspension.

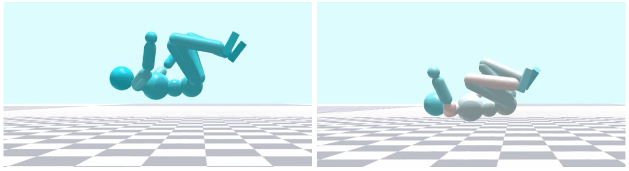

Figure 10: Change of motion style. Left: Non-fatigued backflip; Right: Fatigued backflip. The height of the fatigued backflip is lower and the body is less stretched compared to the non-fatigued backflip.

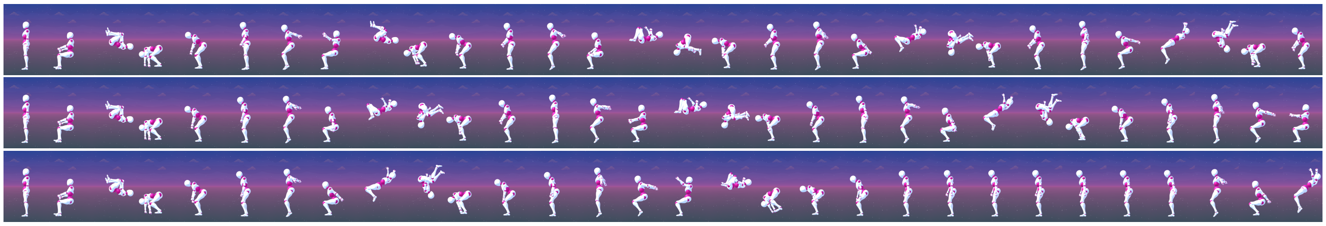

Figure 11: Backflip at different (F, R, r)-settings during inference for our model. *Top:* Non-fatigued backflip at $(0, 0, 0)$. *Middle:* Moderately fatigued backflips at $(1.0, 0.06, 1.0)$. *Bottom:* High fatigued backflips at $(2.0, 0.07, 1.0)$.

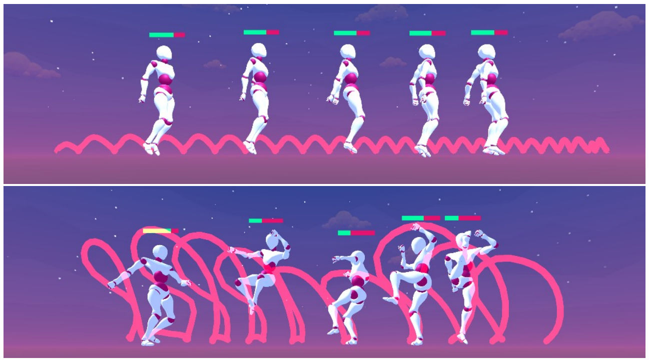

Figure 12: Change of motion performance. Hopping (*top*): The distance between each hop decreases with the accumulation of fatigue. Tornado kick (*bottom*): The range of the kick and the height of the jump both decrease with the accumulation of fatigue. The character then stands and waits for a significantly long time to recover before performing another kick.

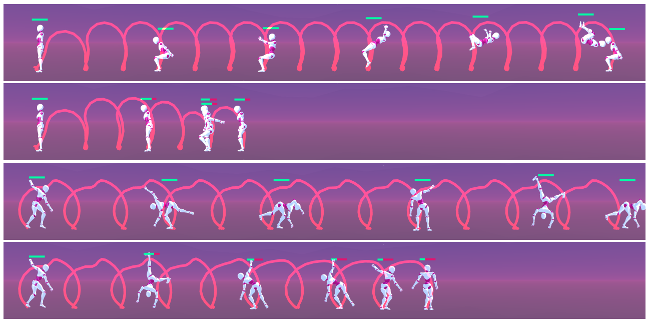

Figure 13: Reduction of number of repetitions. The number of repetitions indicated by the number of loops drops between fatigued and non-fatigued motions during the same amount of time. From top to bottom: non-fatigued backflips, fatigued backflips, non-fatigued cartwheels, fatigued cartwheels.

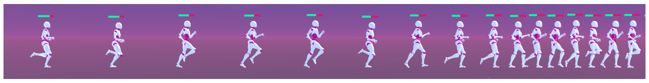

Figure 14: Change of speed. The character becomes slower as it fatigues.

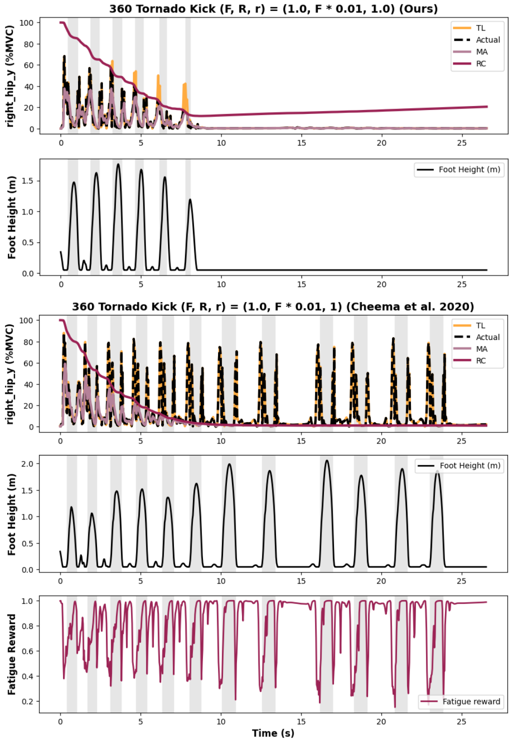

Figure 15: Reward-based fatigue transfer learning [Cheema et al. 2020]. The grey areas indicate a successful tornado kick. *From top to bottom*: %MVC for the foot height for the fatigued training setting (1, 0.01, 1) for our method, as well as the corresponding movement pattern. %MVC for Cheema et al. [2020], as well as the corresponding movement pattern and the fatigue reward *(bottom).* Note how the movement patterns and applied torque levels do not correctly correspond to the level of fatigue.

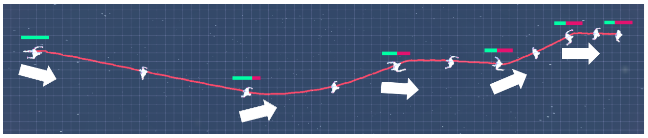

Figure 16: Goal-oriented Walking with Cumulative Fatigue. The character is shown at a 5s interval, while the target direction is changed every 10s as indicated by the arrows. One can observe that the character walks slower while the fatigue accumulates.

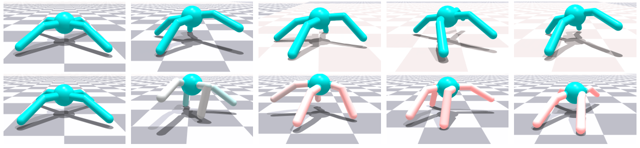

Figure 17: Goal-oriented IsaacGym ‘Ant’. *Top:* Without fatigue. *Bottom:* With fatigued torque bounds. Each picture was captured with a 100-frame interval between them. In a fatigued state the Ant keeps its body low and walks slowly as can be seen in the supplementary video.

A GOAL-ORIENTED TASKS

We conduct a goal-oriented simulation task in which a character moves toward a goal while experiencing growing fatigue. To accomplish this, we employed two distinct environments: 1) The humanoid character utilizing the GAIL-based pre-trained expert, 2) a four-legged spider, specifically the Isaac Gym ‘Ant’ with 8 degrees of freedom (DOFs), representing a character with a distinct morphology that is challenging to replicate through motion capture.

GAIL-Humanoid. We expand upon the state space $\mathbf{s}_t$ outlined in Section 5 of the main document by incorporating a 2D-direction vector representing the character’s intended heading. The reward function in Eq. 8 is then augmented by adding the following heading reward:

$$r_{dir_t} = \left(\left\langle \frac{\Gamma_t}{||\Gamma_t||}, \frac{\hat{\Gamma}}{||\hat{\Gamma}||} \right\rangle + 1 \right) \cdot 0.5$$

With Γ_t describing the current local root x,y-orientation and $\hat{\Gamma}$ the target direction. $\hat{\Gamma}$ is randomized at every environment reset. During inference we set $\hat{\Gamma}$ every 10s, which can be seen in Fig.16.

Four-legged Spider. For this experiment we extend the goal-oriented IsaacGym ‘Ant’-environment with the fatigue-based torque-limits detailed in Section 4.2, and the addition of M_F into state space $\mathbf{s}_t$. Given that we are dealing with a fictional character, we presume a maximum torque limit of 100 Nm for each DOF. The goal of the Ant is to walk towards a specific target. Cumulative fatigue leads to movements that look more natural for the Ant, where it walks slowly on all fours. This can be observed in last two sub-figures in the second row of Fig.17. The original implementation leads instead to speeding and jumping on its hind legs. The results can be better viewed in the supplementary video. In Fig.18 we further show the average speed difference between the two across 64 environments.

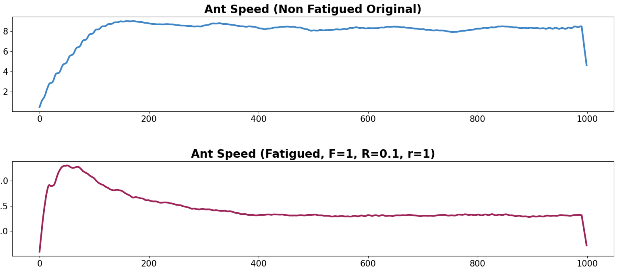

Figure 18: ‘Ant’-Speed averaged over 64 environments. *Top:* Non-fatigued Ants. *Bottom:* Ants with cumulative fatigue.

B LIMITATIONS AND DISCUSSION

While our method for the first time demonstrates plausible fatigue behaviours and emerging resting strategies in the context of full-body character animation, it still exhibits some limitations. Our proposed method is based on the assumption that perceived fatigue can be directly deduced from biomechanical information. However, in reality perceived fatigue can be attributed to a multitude of various factors, such as physiolocal and psychological changes or environmental factors, with fatigue and rest being perceived differently from different individuals [Borg 1982; Frey-Law et al. 2012a; Hincapié-Ramos et al. 2014; Jang et al. 2017; Xia and Frey Law 2008]. In addition, the 3CC model has primarily been only validated on simple isometric tasks. Furthermore, while our method is able to generate unseen motions, they may still result in motions of lower quality compared to methods that completely rely on imitation learning, since we try to generate motions outside of the training distribution. One could improve this by including additional methods to explore the action space further, such as intrinsic motivation or curiosity [Pathak et al. 2017; Yin et al. 2021]. For more naturalness, one could also simulate the motor units MUs in the 3CC model as muscle tendons. However, a key-advantage of the 3CC model for animation is that the MUs can be used as a percentage of maximum voluntary contraction defined as percentage of force or torque, which has also been shown to produce accurate fatigue measures in bio-mechanics literature [Frey Law and Avin 2010; Frey-Law et al. 2012a]. Beyond the general accuracy of fatigue modeling, future directions could include object- or agent-agent interactions [Bae et al. 2023; Hassan et al. 2023; Zhang et al. 2023], as well as the general exploring of motion modeling outside data-distributions similar to Lee et al. [2021a].

Despite these limitations, we believe this work will help pave the way towards developing widely reusable control models for physics-based character animation and biomechanical simulations. As such, our work is of importance for biomechanics and animation researchers and practitioners alike, as it can be employed in various applications such as ergonomics analysis and physical skill training in VR/AR environments, as well as fatigue and stamina animation. In this regard, Digital Human Modeling (DHM) [Demirel and Duffy 2007; Maurya et al. 2019] has been widely used in industrial simulation of workers to perform tasks. In order to achieve biomechanically correct ergonomics assessment, it is necessary

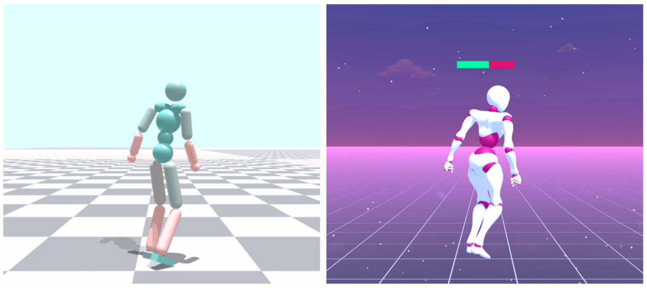

Figure 19: *Left:* Ragdoll simulation model. *Right:* Re-targeted visualization model. Our framework is used to learn fatigued movements for a 28 degrees-of-freedom humanoid character. Pink tints indicates fatigued areas (*left*). The color of the rigid body corresponds to the fatigue of its parent joint (e.g. lower leg corresponds to the knee). The health bar indicates the current average residual capacity (*right*).

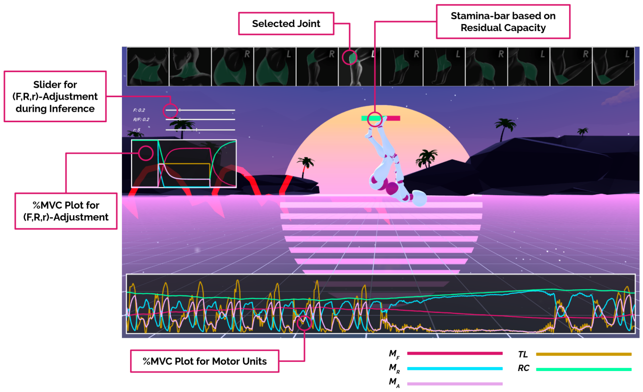
Selected Joint
Stamina-bar based on Residual Capacity
Slider for (F,R,r)-Adjustment during Inference
%MVC Plot for (F,R,r)-Adjustment
%MVC Plot for Motor Units
M_F
M_R
M_A
TL
RC

Figure 20: UI for fatigue modeling. Fatigue synthesis and analysis application based on our method with sliders for setting various (F, R, r)-combinations during inference, and indication of fatigue levels for each joint.

to include an accurate fatigue model of simulated workers. In the scenario of virtual physical skill training, such as virtual sports training, an accurate fatigue model can prevent the users from performing unhealthy movements or getting hurt [Fieraru et al. 2021]. Moreover, our interactive fatigue and rest controls make it possible to easily adapt our method to different fatigue settings during inference in real-time, allowing our approach to be employed in interactive applications such as games or animation tools as can be seen in Fig. 20 in this document and Fig. 1 in the main document. The morphology-agnostic nature of the 3CC-model further allows for the extension to characters of different physiology, and easy extension towards goal-oriented learning.

C ADDITIONAL IMPLEMENTATION DETAILS

For the benefit of the broader reinforcement learning and movement synthesis community, we identify and outline implementation issues and our workarounds used to address them. To leverage the massive GPU parallelization offered by NVIDIA Isaac simulator, we elected to use the carefully tuned default simulation parameters for the `HumanoidAMP` environment. This entails the relatively sparse simulation frequency at 120 Hz and policy query frequency at 30 Hz while using Isaac's native position control mode, which implements stable PD control into the inner loop of the Featherstone articulation program. The inner loop remains inaccessible as Isaac is at a closed-source preview stage as of this writing.

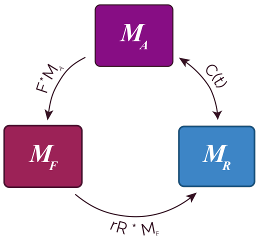

Figure 21: Overview of the Three-Compartment Controller (3CC) model. M_A, M_R and M_F denote the percentage of active, rested and fatigued MUs, respectively. F and R denote the fatigue and recovery coefficients, and $C(t)$ the muscle activation-deactivation drive.

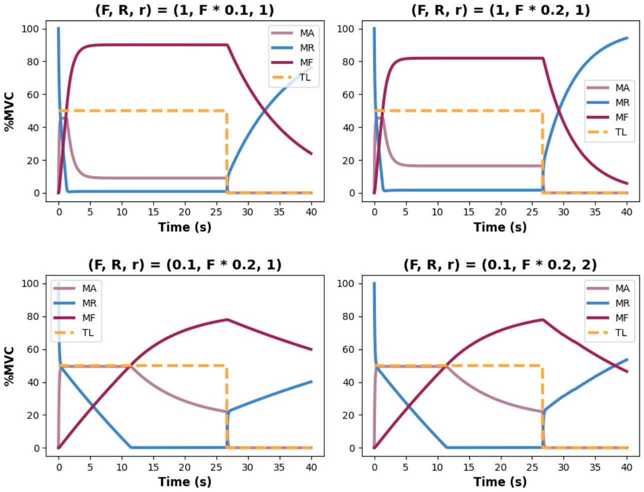

Figure 22: The behaviour of the 3CC model at 50% MVC up until $26s$ with different fatigue F, rest R and rest recovery r rates. Note how the target load TL cannot be held any longer due to fatigue at $\sim 3s$ for $F = 1$ (top two figures), and $\sim 12s$ for $F = 0.1$ (bottom two figures) as M_A decays and is less than TL. A change in F denotes the rate of fatigue. R denotes the limit of fatigued MUs M_F: $R = F \cdot 0.2$ for example, indicates that 20% MVC can be held indefinitely (top right and bottom two figures), as well as overall change of rest. A change in r indicates faster recovery during rest periods, i.e. when $TL = 0$.

Given the limited integration between the constraint solver and the simulator, we were unable to directly obtain accurate actuation readings from the simulator, specifically the torque values in Nm

Algorithm 1 Residual Capacity (RC) per DoF

```
$F \leftarrow$ set the fatigue rate
$R \leftarrow$ set the recovery rate
$r \leftarrow$ set the rest recovery multiplier
$R_r \leftarrow R$ set the rest recovery rate
$L_D \leftarrow$ set the muscle force development factor
$L_R \leftarrow$ set the muscle force relaxation factor
$\Delta t \leftarrow$ interval time between two consecutive simulation steps
$\mathcal{T}_{max} \leftarrow$ set the maximum torque
function 3CC($\mathcal{T}$)
    if at the beginning of an episode then
        if training then
            $M_R \leftarrow Uniform.sample(0, 100)$
            $M_A \leftarrow Uniform.sample(0, 100 - M_R)$
            $M_F \leftarrow 100 - M_R - M_A$
        else
            $M_R \leftarrow 100$
            $M_A \leftarrow 0$
            $M_F \leftarrow 0$
        end if
    end if
    $\mathcal{T} \leftarrow clip(\mathcal{T}, -\mathcal{T}_{max}, \mathcal{T}_{max})$        ▷ Eq. (11)
    $TL \leftarrow \frac{\mathcal{T}}{\mathcal{T}_{max}} \cdot 100$
    if $M_A < TL$ and $M_R > TL - M_A$ then
        $C_t \leftarrow L_D \cdot (TL - M_A)$        ▷ Eq. (3)
    end if
    if $M_A < TL$ and $M_R \leq TL - M_A$ then
        $C_t \leftarrow L_D \cdot M_R$        ▷ Eq. (3)
    end if
    if $M_A \geq TL$ then
        $C_t \leftarrow L_R \cdot (TL - M_A)$        ▷ Eq. (3)
        $R_r \leftarrow r \cdot R$        ▷ Eq. (2)
    end if
    $M_A \leftarrow M_A + \Delta t \cdot (C_t - F \cdot M_A)$        ▷ Eq. (1a)
    $M_R \leftarrow M_R + \Delta t \cdot (-C_t + R_r \cdot M_F)$        ▷ Eq. (1b)
    $M_F \leftarrow M_F + \Delta t \cdot (F \cdot M_A - R_r \cdot M_F)$        ▷ Eq. (1c)
    $RC \leftarrow (100 - M_F)/100$        ▷ Eq. (4)
    return $RC$
end function
```

Algorithm 2 Fatigue Transfer Learning

```
input $\mathcal{M}$ : reference motion
input $\mathcal{T}_{max}$: maximum torque
$D \leftarrow$ initialize discriminator with pre-trained model
$\pi \leftarrow$ initialize policy with pre-trained model
$V \leftarrow$ initialize value function with pre-trained model
$\mathcal{B} \leftarrow \emptyset$ initialize replay buffer
while not done do
    for trajectory $i = 1 \ldots m$ do
        $\tau^i \leftarrow \emptyset$ initialize trajectory
        $s_0 \leftarrow$ sample start state from pre-defined distribution
        for time step $[t = 0, \ldots, T - 1]$ do
            $a_t \leftarrow$ sample action (target joint position) from $\pi$
            $\mathcal{T} \leftarrow PDControl(a_t)$        ▷ Eq. (9)
            for each DoF $d$ do
                $RC_d \leftarrow 3CC(\mathcal{T}_d)$ (see Alg. 1)
                $\widetilde{\mathcal{T}}_{max_d} \leftarrow RC_d \cdot \mathcal{T}_{max_d}$        ▷ Eq. (12)
                $\widetilde{\mathcal{T}}_d \leftarrow clip(\mathcal{T}_d, -\widetilde{\mathcal{T}}_{max_d}, \widetilde{\mathcal{T}}_{max_d})$        ▷ Eq. (13)
            end for
            $s_{t+1} \leftarrow$ get next state by applying $\widetilde{\mathcal{T}}$
            $r_t \leftarrow -\log(1 - D(\Phi(\mathbf{s}_t), \Phi(\mathbf{s}_{t+1})))$        ▷ Eq. (8)
            record $r_t$ in $\tau^i$
        end for
        store $\tau^i$ in $\mathcal{B}$
    end for
    update $\pi$, $V$ and $D$ using data from $\mathcal{B}$ and $\mathcal{M}$
end while
```

Table 2: Custom stiffness and damping parameters used for the humanoid character.

Joint	Stiffness	Damping
Abdomen	120	12
Neck	90	9
Shoulders	100	10
Elbows	110	11
Hips	320	32
Knees	370	37
Ankles	120	12

applied to the motors at each timestep. As an alternative, we considered using the effort mode control mode, but found that it resulted in an unstable simulation at reasonable frequencies. As a result, we decided to use position mode control and estimate the torque values using the explicit PD control formula. We used stiffness and damping parameters modulated as appropriate for each controlled degree of freedom. We are confident that this approximation is reasonable, as the 3CC model used in our research is based on ratios instead of absolute units or scales. For the k_p and k_d parameters at each controlled DoF, we used stiffness and damping parameters modulated as appropriate. We deem this approximation reasonable as the 3CC model is based on ratios instead of absolute units or scales. Recall that the modulated stiffness and damping parameter values are written as $\tilde{k}_{p,t} = \beta_t \dot{k}_p$ and $\tilde{k}_{d,t} = \beta_t \dot{k}_d$, respectively. As such, the intended torque $\mathcal{T}$ and the clipped torque $\tilde{\mathcal{T}}_{max}$ were estimated as

$$\tilde{\mathcal{T}} = \tilde{k}_{p,t} \cdot \left(\hat{\theta}_t - \theta_t\right) - \tilde{k}_{d,t} \cdot \dot{\theta}_t$$

where $\hat{\theta}_t$ is the PD target angles corresponding to the policy action a_t, θ_t is the current DoF angles, and $\dot{\theta}_t$ is the current DoF angular velocities, all at timestep t. To implement the torque limiting described in the main paper, in case the estimated intended torque $\mathcal{T}$ is above the the effective ceiling torque $\tilde{\mathcal{T}}_{max}$, we compute the ratio between them, which is then multiplied to the intended torque to implement the clipping, i.e.,

$$\tilde{\mathcal{T}} = \begin{cases} \mathcal{T} \cdot \frac{\tilde{\mathcal{T}}_{max}}{\mathcal{T}} \\ \mathcal{T} & \text{otherwise.} \end{cases}$$

We set the simulator's joint stiffness and damping values to $\tilde{k}_{p,t}$ and $\tilde{k}_{d,t}$, respectively.

Table 3 reports additional hyperparameters for training our approach such as learning rate and batch size. Table 4 reports the found maximum torque bounds. We report bounds for each action and joint. Note that those bounds are found automatically during our proposed pre-training.

Table 3: Additional PPO-Hyperparameter Settings.

PPO-Hyperparameters	
Parameter	**Value**
Horizon Length	16
Minibatch Size	4096
Batch Size	512
KL Coeff.	0.008
γ Discount	0.99
GAE(γ)	0.95
TD(γ)	0.95
Adam stepsize	5×10^{-5}

Table 4: Found maximum constant torque bounds for each task during pre-training. Note that z denotes the up-axis. Symmetric joints present on the left and right side of the body use the same values. Bold values highlight the maximum value across motions, hence the denominators of TL value calculation.

Maximum constant torque bounds found during Training					
DoF	**Run**	**Hop**	**Cartwheel**	**Backflip**	**Kick**
$Abdomen_x$	45	67	163	**370.27**	290
$Abdomen_y$	37	197	234	**382.34**	181
$Abdomen_z$	57	98	155	**240.88**	234
$Neck_x$	14	26	59	21	**78.61**
$Neck_y$	27	**160.34**	133	77	69
$Neck_z$	15	16	62	**63.66**	47
$Shoulder_x$	20	64	149	**177.67**	143
$Shoulder_y$	35	98	90	**362.03**	171
$Shoulder_z$	13	56	116	**306.93**	173
$Elbow$	22	107	80	91	**136.1**
Hip_x	150	394	580	143	**598.84**
Hip_y	203	421	**796.42**	388	691
Hip_z	79	101	247	135	**373.14**
$Knee$	421	741	496	340	**809.59**
$Ankle_x$	23	92	99	73	**138.58**
$Ankle_y$	228	207	120	407	**451.8**
$Ankle_z$	38	70	115	90	**201.1**