Learn to Sprint 100 Metres

Yan Ma
mayan20@fudan.edu.cn

Abstract

As a fanatic of *track and field*, *reinforcement learning* and *computer animation*, I aspire to control simulated characters in the virtual world to complete various athletic sports. This research proposal contrives a staged framework and demonstrates how to learn intricate 100-meter sprint. The first section introduces the motivation, background, significance and technical difficulties of the study. The second section presents preliminary of modeling the 100m sprint as a paradigm of reinforcement learning, including the state and action space design, etc. The third section elaborates on a set of deliverable solutions, breaking through the learning of the 100-meter dash one by one in stages. The last section discusses related work.

1 Introduction

The study of physics-based controllers for character animation is a long-term research problem. Researchers have proposed various approaches for learning motion skills. In the pursuit of faster, higher and stronger sports, this proposal specializes in driving a physically simulated character to sprint 100 meters [1] like a track and field athlete. One hundred metres is a trinity of strength, skill and speed, representing the exploration and pursuit of human limits. It is disparate from running, which has been extensively studied as a fundamental human skill and served as a benchmark for evaluating control algorithms [2]. Compared with running, the discrepancy and difficulty of 100-meter dash primarily lie in three aspects: 1) More speedy. Sprinting speed is much faster than the average running speed of human beings. The faster the body travels, the higher the demand for stability and balance. 2) More synergy. Sprinting posture reveals the beauty of strength, which requires synergy all over the body, rather than running, which principally focuses on the motion of legs. 3) more energy efficient. After more than 100 years of development, the modern sprint pose has formed a set of complex and specific techniques which can remarkably reduce energy loss and improve the pace of movement. These three points make learning 100 metres a nuisance. As the jewel in the crown of athletics, the acquisition of 100-meter dash can vastly enhance the motor ability of characters which implies that simulated agents will not only be able to grasp the basic human motor skills (e.g. walking and flipping) but also have the ability to learn athletic sports that only the top humans can touch. It will be a breakthrough in character control.

So, how to learn the 100-meter dash? An obvious solution is to track motion capture data like DeepMimic [3]. However, the difficulty is that the available mocap data for sprinting is barren. Moreover, we desire to learn rather than imitate the whole process of sprinting. To that end, We propose a learning-based framework that divides the 100m sprint into four phases: *Start*, *Acceleration*, *Maintain Top Speed* and *Dip Finish* and break them down in turn. Specifically, a dataset for 3D human pose estimation is compiled from the downloaded 100m race videos, which is the preliminary work and accessory throughout the entire learning process. For *Start*, We adopt Imitation Learning (IL) to reproduce a plausible departure. For *Acceleration* and *Maintain Top Speed*, we embrace Reinforcement Learning (RL) combined with motion primitives captured by Auto-Encoder(AE) to enable a realistic sprinting pose. For *Dip Finish*, we employ Evolutionary Algorithm (EA) to explore a suitable finish strategy. Through a series of training, the character agent is eventually capable of sprinting 100 meters like athletes.

Figure 1: The men's 100m final competition at the 2012 Summer Olympics in London.

2 Preliminary

The task of controlling a character agent in physics simulation to finish 100 metres can be formulated as a Markov decision process (MDP), which is typically stereotyped as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma)$ of states, actions, transition dynamics, a reward function and a discount factor. The character agent leverages a policy $\pi(a_t|s_t)$ which maps the current state $s_t \in \mathcal{S}$ to the desired action $a_t \in \mathcal{A}$. At each timestep t, the character iteratively performs an output action a_t from policy π and carries out the transition to the next state s_{t+1} according to the transition dynamics $\mathcal{T}(s_{t+1}|s_t, a_t)$. The goal of the controlling task is to discover an optimal policy π_* which maximizes the long-term return $J(\theta) = \mathbb{E}_\pi[\sum_t \gamma^t r_t]$.

State. The state s is typically a hand-crafted configuration which specifies features of the character's body. The features are a set of pose (position and rotation) q_t of each link and joint linear and angular velocity $\dot{q}_t$. In addition, we also add the body's center of mass c_t, which athletes will control at different stages of sprint to obtain momentum in the direction. More importantly, a scalar variable z_t is included, which refers to the number of steps that have been physically advanced (i.e. the number of alternating legs). The justification is that athletes typically finish 100 meters with a relatively fixed number of steps and judge what action should be taken by the number of steps they have already forwarded. Therefore, we introduce a normalized step number z rather than a phase variable prevalent in prior work [3].

Action. The common definition of action a in RL community is the torque τ_t to be applied at each joint. When dealing with character agents with high degrees of freedom (i.e. 34 DoF Humanoid), torque output may lead to some artifacts such as jitter and overactivity. We adopt prior work [3, 4, 5] and actuate the character by vanilla or stable PD-controller [6] positioned at each joint. Specifically, each action a_t specifies target rotations, which is parameterized by the quaternion form for spherical joints and a scalar angle for revolute joints. In contrast to torque-based controller, the PD controller frees RL algorithms from the low-level control details (e.g. damping or other manually-specified gains) and can speed up learning. It is worth mentioning that there are several alternative options for character controller. [7] uses the mixture of torque-based and PD controllers as the action space. Furthermore, [8] explicitly supports the use of torque-based controllers for character control. In principle, the torque-based controller performs better in the case of large-scale motion dataset as empirically proven in [8]. Given the barren motion data of sprint, we opt to exert the character with PD-controller.

Reward. This section primarily introduces the basic reward design r_{base} composed of three items throughout the learning process. The additional reward term for each sprint part will be detailed in Section 3. 1) Forward reward r_f. The faster the character moves forward, the higher the reward. The character's speed is derived by dividing the root position difference between two adjacent frames by the timestep. 2) Survival reward r_s. It will be available while the agent does not fall down. 3) Penalty for deviation r_d. Since 100 metres is forward locomotion alongside the x-axis in global coordinate, punishment will be imposed once deviated, and the magnitude depends on the offset. The consolidation of these three items specifies the primary reward $r_{base} = r_f + r_s + r_d$. Additive terms will be integrated into the total reward through summation.

Initial State. The effect of the initial state on learning has been extensively studied in previous work [3, 9, 10, 5, 11]. A decent initial state can accelerate skill learning and enhance robustness, and vice versa. For instance, [5] enables the agent to exploit diverse takeoff strategies for the high jump by searching various and excellent initial states. However, 100 metres is an athletic sport with a relatively fixed initial pose, which signifies proficient sprinters only need a few starting techniques and reuse them in each competition. Therefore, we will not conduct diverse discovery of the initial pose in our case. Instead, we will collect the various starting postures from sprint race videos and assign a specific one whenever the task is reset. Incidentally, Domain Randomization is applied during learning process, which is a popular technique to enhance robustness. We randomly vary wind speed (the key factor affecting sprint performance) and runway friction within a predefined range, so that the controller is able to finish 100-meter dash with quality and quantity under various wind speeds and pitches.

3 Methodolody

As mentioned above, 100 metres can be composed of four parts: *Start*, *Acceleration*, *Maintain Top Speed* and *Dip Finish*. The following sections will describe the detailed methods used in each part respectively. Before that, we opt to create a sprint race dataset in advance, in that 100 metres has several characteristics that distinguish it from running. For example, the proper start, leg raise, foot placement and finish line postures of athletes can exert body's potential to enhance speed, which are the most suitable sprinting pose for humanoid evolved over the past hundred years. However, these particular postures are difficult to learn from scratch through a hand-crafted reward function. Therefore, an athlete pose dataset is a necessity to procure natural and plausible postures for the character. Firstly, we collect match videos from *Diamond League*, *World Athletics Championships* and *Olympic Games*. Then, we extract 3D skeletal poses from race videos using pose estimation, which will be served as the reference data. There are a bunch of 3D pose estimation techniques [9, 12, 13] can help to reconstruct the kinematic or physically-grounded estimated motion data from monocular videos. The dataset will be integrated into each learning part and improve the motion quality.

3.1 Start

The purpose of *Start* is to make the body quickly get rid of the stationary state and obtain the starting speed. Specifically, the athletes utilize the starting blocks and conduct crouch starts. A decent start can help athletes accumulate energy for acceleration. We plan to learn *Start* with Imitation Learning (IL) approach and the justification is twofold. First and foremost, the start technique in modern 100 metres is relatively fixed and the technical nuances among athletes only lies in trifles (e.g. the starting foot or crook of elbow). Meanwhile, the start stage merely lasts about 2.5 meters (i.e. the first step after the departure). This means that the agent simply needs to perform an almost invariant movement within a short horizon, which is what IL is good at. Therefore, IL is a plausible choice for learning *Start* in light of the stereotypical technique and short horizon.

We extract all poses from departure to first foot landing in the sprint race dataset, which will be utilized as references and encourage the character to reproduce *Start* process via imitation. Simple Behavioral Cloning (BC) or Generative Adversarial Imitation Learning (GAIL) [14] are both viable. It depends on the actual performance. We keep the initial pose of the character consistent with that in the current reference trajectory and periodically alter a new reference. The agent will finally empower *Start* under various departures. We sample a batch of available initial poses in the dataset to detect the controller performance. The imitation loss $r_{imitation}$ is integrated into the basic reward r_{basic} as an additional term with a weight coefficient λ_i. The character agent is considered to have completed *Start* when one foot hits the ground, and the posture at that time will be adopted as the initial state for the next stage of learning.

3.2 Acceleration & Maintain Top Speed

Acceleration and *Maintain Top Speed* are two adjacent stages in 100 metres. Generally, the duration of *Acceleration* ranges from 20 to 40 meters (younger is generally shorter while elite is generally longer). During this period, athletes maintain core stability and avoid the shake of trunk. At the same time, the upper part of the body leans forward to 45 degrees. As the speed increases, athletes gradually rise the centre of mass and transition to the next stage. *Maintain Top Speed* is the stage of the longest distance and fastest speed in the sprint, with the purpose of exerting and maintaining the highest velocity. During this stage, athletes maintain high center of mass and core stability until close to the finish line.

We intend to treat these two stages as a whole and leverage RL algorithms to train the corresponding policy, in that both phases are similar in leg swing. The difference only lies in the posture of the upper body. In addition, the switch between the two stages is the rise of the body's center of mass, which has a great impact on movement and should be chosen by the athletes themselves. We desire that the agent can learn the altered timing by itself instead of setting the time point manually. Therefore, the two stages are considered as one and accompanied by a self-learning process of upper body upright. In order to depict the transformation of the center of mass, a penalty term r_{mass} is constructed and added to the reward function, which punishes the gap between the center of mass in the current and the upright posture.

During *Acceleration* and *Maintain Top Speed*, athletes drive their arms and legs forward. Their arms swing back and forth briskly, driving the motion of their legs while maintaining body balance. We divide the learning process into two parts: 1) learn the sprint posture; 2) learn motion symmetry.

Learn the sprint posture. The basic reward function r_{basic} described in Section 2 can make the character move forward rapidly. However, it is not easy to reproduce the realistic sprint style (e.g., folding of legs and swing of arms). Therefore, additional techniques are required to regularize the sprint pose. We reaffirm that our destination is to learn the sprint rather than blindly imitating any athlete. To that end, we take no account of utilizing the sprint postures as the tracking reference but resolve to maneuver the latent space of posture trajectory via variational auto-encoder(VAE) [15] to produce a natural pose. Inspired by contrastive learning [16, 17, 18], we contrive a deliverable solution as follows. First, we align all the collected sprint trajectories according to the angle between the two thighs and truncate the redundant parts. Then, we pretrain a VAE Φ_1 with aligned posture trajectories. It captures the low-dimensional manifold of natural sprint posture from the athlete, which is called sport primitive z in this proposal. Meanwhile, we align the pose trajectory τ generated by the character with a batch of trajectories $\{\tau_{1...n}\}$ sampled from the sprint dataset and initialize another VAE Φ_2 trained by aligned τ. At last, we freeze the parameters of Φ_1 and minimize the distance (e.g., KL divergence) of sport primitive between τ and $\{\tau_{1...n}\}$. Repeat the last two steps until the maximum training steps. The merit of sport primitive is the implicit stipulation for naturalistic posture, which alleviates several notorious artifacts generated by RL algorithms (e.g., the frantic jitter) and may enable characters to comprehend the internal mechanism of the sprint rather than imitate and reproduce. The approximation of sport primitive between Φ_1 and Φ_2 will improve the similarity of the character's posture and athletes' posture in latent space and ultimately assist the character in achieving an elegant sprint posture.

Learn motion symmetry. There are two kinds of symmetry in sprinting: One is that the motion of legs and arms is diagonal, and the other is that the posture in odd and even steps is mirror-symmetrical. As illustrated in prior work [19, 20], motion symmetry is essential for the production of high-quality and naturalistic motion posture. In order to achieve symmetry in 100 metres, we devise an auxiliary loss term $r_{symmetry}$ to promote the diagonal arm and thigh stretching simultaneously and stimulate the compatibility between the current pose and mirrored previous pose. The dual integration of the sport primitive captured by auto-encoder and the designed loss for symmetry enables the character agent to sprint forward efficiently and exquisitely until the finish line.

3.3 Dip Finish

The *Dip Finish* is the final stage, and athletes lean forward quickly and timely when they are one to two steps away from the finish line. The trunk will touch the line faster than the normal sprint posture through the dip finish to elevate the final time. However, an excellent reference for dip finish is still being determined. Athletes should choose the appropriate moment to tilt their bodies according to competition circumstances. Meanwhile, dip finish is unsuitable for RL paradigm, as it is a decision made at a single time step. More importantly, judging whether a dip finish is good or not is nontrivial, and the criterion is just the final time. Evolutionary Algorithms (EA) can rise to the occasion for this tricky case, providing a black-box solution. We abstract the dip finish into two coefficients: the dipping moment t_{dip} and the tilt angle θ_{dip}, representing one θ_{dip} degree rotation exerted on the root at a specified timestep t_{dip}. The dipping moment t_{dip} is a scalar that varies from one to five meters from the finish line. The initial value is five (corresponds to the two-step length). The tilt angle θ_{dip} is a degree in a range of 0 to 90, and the initial value is 30 degrees. Then, a multivariate Gaussian distribution Ξ is constructed with these two coefficients as the mean value, from which a batch of moments and angles are sampled periodically. We exert dip finish according to these sampled moments and angles and record the final results as the fitness score. Afterward, we update Ξ through Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [21] to make its mean value move towards the area with higher results. Repeat this process until it converges, and the final Gaussian represents the strategy for dip finish. When the dash reaches the selected dipping moment, the character's upper body bends at the specified angle to complete *Dip Finish*.

4 Related Work

4.1 Physics-based Character Animation

Developing physics-based controllers for character animation is a long-term research challenge. Early work utilizes hand-crafted finite state machine (FSM) or proportional-derivative (PD) as the character controller [22, 23, 24]. Optimal control approaches prevalent in robotics community (e.g., model predictive control, linear quadratic regulator) have been widely adopted in character animation for a time as well [25, 26, 27, 28]. We refer readers to a survey paper for more antique approaches [29]. The following mainly focuses on learning-based techniques used in modern character animation, which has become the mainstream in the era of Deep Learning.

[30] comprehensively reviewed a series of learning-based methods in character animation. [31] exhibited policy gradient methods with a naive forward reward is enough to learn various motor skills although the final result needs further realism. [32] adopted a two-level hierarchical control framework trained with RL. The high-level controller conducts a long-horizon plan according to terrain conditions. The low-level controller is responsible for short-horizon gait control. Although RL-trained neural controllers have made brilliant achievements in learning locomotion skills, they often produce unnatural motion postures. The justification is the difficulty of depicting a naturalistic pose via an artificially defined reward function. Some research endeavors [19, 20] have designed heuristic symmetry constraints to incentivize the symmetry of motion implicitly. Inspired by prior work, we contrived a set of symmetry rules compatible with 100 metres, including the diagonal hand and foot stalemate and the mirror symmetry of adjacent motions.

The long-term motivation to produce realistic motions spurs the adoption of Imitation Learning (IL) in character control. [33, 34, 35, 36, 3] The imitation is implemented through tracking reference clips to minimize the error between the simulated character's current pose and the reference's pose. The main limitation of these tracking-based methods lies in the difficulty of learning large-scale multi-skills since they rely on the phase variable, which varies among each reference clip. Then, researchers introduced Adversarial Imitation Learning (AIL) [37, 14] to deal with the above shortcomings. AIL trains a discriminator to distinguish the behavior depicted in demonstrations from any other behaviors. The control policy should try its best to imitate the demonstration and trick the discriminator. Through adversarial training, a control strategy can master multiple skills simultaneously from a large-scale unstructured database containing diverse mocap clips [38, 39, 40, 41]. Our method utilizes IL to reproduce diverse *start* processes. For *Acceleration* and *Maintain Top Speed*, we take no account of imitation with reference motions but to learn the practical sprint skill suitable for characters by themselves.

In order to learn a diverse array of motor skills, generative models [15, 42] are alternative techniques. Such methods typically learn or predefine a compact latent space (called motion primitive or skill embedding) that represents a variety of motions [43, 44, 45, 41, 46]. The samples from latent space can be decoded as specific skill motions for simulated characters. However, our approach is devoted to regularizing the realism of sprint posture through compact latent space rather than learning large-scale skills. Similar to the spirit of our work spirit [5], It pre-trained a β-VAE [47] on a large-scale unstructured mocap database as a representation of realistic poses. The control policy outputs motion offsets d_{offset} and latent poses d_{latent}. The pretrained β-VAE will decode d_{latent} into natural postures and apply motion offsets the VAE output poses as the final PD target. We also adopt β-VAE as the extractor of motion primitives. However, we do not employ a large-scale unstructured mocap database but apply 3D pose estimation to collect the sprint pose dataset from 100m race videos. The large-scale mocap database may impede the learning process since there are plenty of postures significantly different from the naturalistic sprint poses Additionally, we employ two VAE models one is pretrained by the collected sprint pos database. The other is trained from scratch during *Acceleration* and *Maintain Top Speed*. By minimizing the latent space error of sprint postures between the character and athletes, we can promote the production of naturalistic sprint posture.

Black box optimization, such as evolutionary algorithms, is prevalent in nondifferentiable issues. [48] employs NEAT [49], a neural evolutionary technique, to realize the learning of bicycle stunt controllers. With the improvement of computing power, evolutionary algorithms with high parallelism are competent to handle high-dimensional control tasks. [50]. Due to the low sample efficiency, black box optimization has not become the mainstream method in deep learning. Nevertheless, as a surprise component, it can be competent for the nondifferentiable subproblem in character control, for instance, hyper-parameter optimization [51], optimal pose optimization [52]. We adopt CMA-ES [21] to search the plausible moment and tile angle for *Dip Finish*.

4.2 Athletic Skill Learning

Simulating human motion is a long-term intersection of biomechanics and computer graphics. Limited by our knowledge, we mainly refer to computer animation work here. Athletic skills, especially track and field, are the summit of motion skills and require extreme mechanics and explosiveness in a long horizon. Learning sophisticated athletic skills is a great leap forward in character control, which indicates the simulated character can reach the top motion ability of human beings and has the potential to guide human motion in turn. The Olympic Games for simulation agents may even emerge as the times require in the near future. As early as the last century, researchers have attempted to simulate athletic skills like running, cycling and vaulting via a hand-crafted finite state machine [53, 54, 55]. [56] designed user interfaces and implemented interactive control for physics-based platform diving and snowboarding. [48] simulated a humanoid cyclist performing bike stunts via derivative-free evolutionary algorithms. However, prior work demands trial and error and tedious offline manual tuning. In the era of Deep Learning, neural controllers trained by learning-based methods have gradually become mainstream to produce athlete skills, such as rock climbing [57], basketball dribbling [58], figure skating [59], parkour [60], high jump [5], boxing and fencing [61], soccer juggling [62] and so on. These research efforts are devoted to reproducing the athletic motion clips(e.g., bouncing football and dribbling) with mocap data or videos rather than completing a whole game. Moreover, few research endeavors are involved in track and field, which requires high-precision techniques. We pursue a complete 100-meter dash [1] without the support of mocap data, which is more challenging than previous work.

4.3 Learn Physics-based Skills from Videos

Many character control methods adopt motion capture data as a reference for tracking. However, given the professionalism of 100 metres, high-quality motion capture data is barren. An alternative technique is to exert human kinematics or physical pose estimation from videos. Kinematics-based and physics-based pose estimation is an important research area in computer vision. There are countless research works emerging [63, 64, 13, 65]. This proposal focuses on physics-based character animation, so we mainly introduce its intersection with pose estimation. For detailed posture estimation techniques, we refer the reader to a survey paper [66]. [67, 68] perform pose estimation from ego-based videos and apply it to motion tasks. However, for 100 metres, ego-based video is almost extinct. Therefore, we specialize in human pose estimation from monocular videos. SFV [9] is a video variant of DeepMimic, which employs a 3D pose estimator regularized by a 2D pose estimator to reconstruct the kinematic pose from monocular video clips. The reconstructed pose trajectories are served as references to learn various acrobatic skills (e.g., forward roll). Inspired by SFV, [59] applied trajectory optimization to the 3D character pose estimated from YouTube videos and realized the figure skating skill learning. [12] accurately implemented the monocular physics-based 3D human pose estimation by integrating images-based kinesthetic input and RL-based character control. It demonstrated the motion skills such as walking, jumping, opening and closing. [60] pointed out that SFV can only be available under blank background and static camera view. It introduced an additional depth contact estimator to realize dance, gymnastics and parkour skills under various scenes and dynamic camera views. Despite the increasing sophistication of human posture estimation techniques, it still struggles to match high-quality mocap data. Due to the gap between pose estimation and motion capture, most previous work reproduces basic skills from videos in a short horizon. Tracking directly with estimated poses may cause an accumulation of errors leading to unstable sprints for an intricate long-horizon athletic skill like 100 metres. Therefore, we employ off-the-shelf techniques to obtain sprint postures and implicitly promote the realism of sprint as described in Section 3.2.

References

[1] Olympics. The most exciting 100m races in Olympic history! `https://www.youtube.com/watch?v=u_f0697yLBI`, 2015.

[2] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[3] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[4] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and KangKang Yin. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021.

[6] Jie Tan, Karen Liu, and Greg Turk. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, 2011.

[7] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th annual international conference on motion, interaction, and games*, pages 1–10, 2018.

[8] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion. *arXiv preprint arXiv:2011.15119*, 2020.

[9] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6):1–14, 2018.

[10] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)*, 38(6):1–11, 2019.

[11] Tianxin Tao, Matthew Wilson, Ruiyu Gou, and Michiel van de Panne. Learning to get up. *arXiv preprint arXiv:2205.00307*, 2022.

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

[13] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *Acm Transactions On Graphics (TOG)*, 39(4):82–1, 2020.

[14] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[17] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

[18] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[19] Wenhao Yu, Greg Turk, and C Karen Liu. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[20] Farzad Adbolhosseini, Hung Yu Ling, Zhaoming Xie, Xue Bin Peng, and Michiel van de Panne. On learning symmetric locomotion. In *Proc. ACM SIGGRAPH Motion, Interaction, and Games (MIG 2019)*, 2019.

[21] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

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

[23] KangKang Yin, Kevin Loken, and Michiel Van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)*, 26(3):105–es, 2007.

[24] Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. Generalized biped walking control. *ACM Transactions On Graphics (TOG)*, 29(4):1–9, 2010.

[25] Marco da Silva, Yeuhi Abe, and Jovan Popović. Interactive simulation of stylized human locomotion. *ACM Transactions On Graphics (TOG)*, 27(3):1–10, 2008.

[26] Marco Da Silva, Yeuhi Abe, and Jovan Popović. Simulation of human motion data using short-horizon model-predictive control. In *Computer Graphics Forum*, pages 371–380, 2008.

[27] Uldarico Muico, Jovan Popović, and Zoran Popović. Composite control of physically simulated characters. *ACM Transactions on Graphics (TOG)*, 30(3):1–11, 2011.

[28] Igor Mordatch, Martin de Lasa, and Aaron Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics (TOG)*, 29(4), 2010.

[29] Thomas Geijtenbeek and Nicolas Pronost. Interactive character animation using simulated physics: A state-of-the-art review. In *Computer graphics forum*, pages 2492–2515, 2012.

[30] Ariel Kwiatkowski, Eduardo Alvarado, Vicky Kalogeiton, C Karen Liu, Julien Pettré, Michiel van de Panne, and Marie-Paule Cani. A survey on reinforcement learning methods in character animation. In *Computer Graphics Forum*, pages 613–639, 2022.

[31] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[32] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[33] Yoonsang Lee, Sungeun Kim, and Jehee Lee. Data-driven biped control. *ACM Transactions on Graphics (TOG)*, 29(4), 2010.

[34] Kwang Won Sok, Manmyung Kim, and Jehee Lee. Simulating biped behaviors from human motion data. *ACM Transactions on Graphics (TOG)*, 26(3):107–es, 2007.

[35] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. *ACM Transctions on Graphics*, 29(4):Article 128, 2010.

[36] Libin Liu, Michiel van de Panne, and KangKang Yin. Guided learning of control graphs for physics-based characters. *ACM Transactions on Graphics*, 35(3), 2016.

[37] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438, 2008.

[38] Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017.

[39] Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems*, 30, 2017.

[40] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021.

[41] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *arXiv preprint arXiv:2205.01906*, 2022.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[43] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711*, 2018.

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

[45] Ying-Sheng Luo, Jonathan Hans Soeseno, Trista Pei-Chun Chen, and Wei-Chao Chen. Carl: Controllable agent with reinforcement learning for quadruped locomotion. *ACM Transactions on Graphics (TOG)*, 39(4):38–1, 2020.

[46] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *arXiv preprint arXiv:2210.06063*, 2022.

[47] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[48] Jie Tan, Yuting Gu, C Karen Liu, and Greg Turk. Learning bicycle stunts. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[49] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[50] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[51] Zeshi Yang and Zhiqi Yin. Efficient hyperparameter optimization for physics-based character animation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(1):1–19, 2021.

[52] Zeshi Yang, Kangkang Yin, and Libin Liu. Learning to use chopsticks in diverse gripping styles. *ACM Transactions on Graphics (TOG)*, 41(4):1–17, 2022.

[53] Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78, 1995.

[54] Wayne L Wooten and Jessica K Hodgins. Animation of human diving. In *Computer Graphics Forum*, pages 3–13, 1996.

[55] Wayne Lewis Wooten. *Simulation of leaping, tumbling, landing, and balancing humans*. Georgia Institute of Technology, 1998.

[56] Peng Zhao and Michiel van de Panne. User interfaces for interactive control of physics-based 3d characters. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 87–94, 2005.

[57] Kourosh Naderi, Amin Babadi, and Perttu Hämäläinen. Learning physically based humanoid climbing movements. In *Computer Graphics Forum*, pages 69–80, 2018.

[58] Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[59] Ri Yu, Hwangpil Park, and Jehee Lee. Figure skating simulation from video. In *Computer graphics forum*, pages 225–234, 2019.

[60] Ri Yu, Hwangpil Park, and Jehee Lee. Human dynamics from monocular video with dynamic camera movements. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021.

[61] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[62] Zhaoming Xie, Sebastian Starke, Hung Yu Ling, and Michiel van de Panne. Learning soccer juggling skills with layer-wise mixture-of-experts. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.

[63] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. Physics-based human motion estimation and synthesis from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11532–11541, 2021.

[64] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[65] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)*, 39(6):1–16, 2020.

[66] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019.

[67] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 735–750, 2018.

[68] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10082–10092, 2019.