3D Ego-Pose Estimation via Imitation Learning

Ye Yuan[0000−0001−5316−6002] and Kris Kitani[0000−0002−9389−4060]

Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract. Ego-pose estimation, *i.e.*, estimating a person's 3D pose with a single wearable camera, has many potential applications in activity monitoring. For these applications, both accurate and physically plausible estimates are desired, with the latter often overlooked by existing work. Traditional computer vision-based approaches using temporal smoothing only take into account the kinematics of the motion without considering the physics that underlies the dynamics of motion, which leads to pose estimates that are physically invalid. Motivated by this, we propose a novel control-based approach to model human motion with physics simulation and use imitation learning to learn a video-conditioned control policy for ego-pose estimation. Our imitation learning framework allows us to perform domain adaption to transfer our policy trained on simulation data to real-world data. Our experiments with real egocentric videos show that our method can estimate both accurate and physically plausible 3D ego-pose sequences without observing the cameras wearer's body.

Keywords: first-person vision, pose estimation, imitation learning

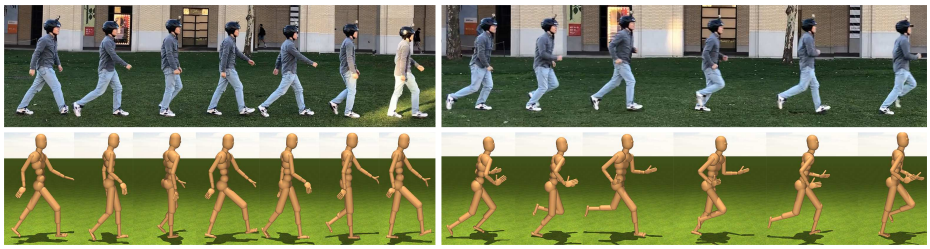

Fig. 1. Our 3D ego-pose estimation results using egocentric videos.

1 Introduction

Our task is to use a single head-mounted wearable camera to estimate the 3D body pose sequence of the camera wearer, such that the estimated motion sequence obeys basic rules of physics (*e.g.*, joint limits are observed, feet contact

the ground, motion conserves momentum). Employing a single wearable camera to estimate the pose of the camera wearer is useful for many applications. In medical monitoring, the inferred pose sequence can help doctors diagnose patients' condition during motor rehabilitation or general activity monitoring. For athletes, egocentric pose estimation provides motion feedback without instrumenting the environment with cameras, which may be impractical for sports like marathon running or cross-country skiing. In virtual reality games, the headset wearer's poses can be reproduced in the virtual environment to create a better multi-player gaming experience without additional sensors. In many applications, accurate and *physically-valid* pose sequences are desired.

However, estimating physically-valid 3D body poses from egocentric videos is challenging. First, egocentric cameras typically face forward and have almost no view of the camera wearer's body. The task of estimating 3D pose is underconstrained as the video only encodes information about the position and orientation of the camera's viewpoint. Second, with a single wearable camera, we have no access to the forces being applied to the body, such as joint torques or ground contact forces. Without observations of these forces, it is very difficult to learn the relationship between camera-based motion features and body pose using physics simulation in a data-driven way. Most traditional approaches to human pose estimation in computer vision side-step the issue of physics completely by focusing primarily on the kinematics of human motion. Unfortunately, this can sometimes result in awkward pose estimates that allow the body to float in the air or joints to flex beyond what is physically possible, which makes it difficult to use for motion analysis applications. New technical approaches are needed to tackle these challenges of generating physically-valid 3D body poses from egocentric video.

In light of these challenges, we take a radical departure from the kinematics-based representation traditionally used in computer vision towards a control-based representation of humanoid motion commonly used in robotics. In the traditional kinematics-based representation used for pose estimation from videos, a human pose sequence is typically modeled as a sequence of poses $\{p_1, \ldots, p_T\}$. It is common to use a temporal sequence model (*e.g.*, hidden Markov model, linear chain CRF, recurrent neural network) where the estimate of each pose p_t is conditioned on image evidence I_t and a prior pose p_{t-1} (or some sufficient statistics of the past, *e.g.*, hidden layer in the case of RNN). While it is often sufficient to reason only about the kinematics of the pose sequence for pose estimation, when one would like to evaluate the physical validity of the sequence, it becomes necessary to understand the *control input* that has generated each pose transition. In other words, we must make explicit the torque (control input) that is applied to every joint to move a person from pose p_t to p_{t+1}. Under a control-based method, a human pose sequence needs to be described by a sequence of states and actions (control inputs) $\{s_1, a_1, s_2, a_2, \ldots, s_T\}$ where state s_t contains both the pose p_t and velocity v_t of the human. A control-based model explicitly takes into account the control input sequence and learns a control policy $\pi(a|s)$, that maps states to actions for optimal control. Making explicit

the control input is essential for generating a state sequence based on the laws of physics.

The use of a control-based method requires access to interaction with real-world physics or in our scenario, a physics simulator. The use of a physics simulator for learning a control policy provides two major advantages. First, the physical properties of the virtual humanoid, such as joint actuation limits and range limits, used in the simulator serve as a gating mechanism to constrain the learning process to generate actions that are humanly possible. Second, the physical constraints of the simulation environment ensure that only physically-valid pose sequences are estimated such that the feet will not penetrate the ground or slip during contact. Within the confines of the physic simulator, the goal of control policy learning is to learn a virtual humanoid policy that maps the current state (pose and velocity, optionally egocentric video) to an action (joint torques). Formally, we frame first-person pose estimation as a sequential decision process using a Markov decision process (MDP). The state of the MDP is the state of our humanoid model defined in terms of joint positions, joint velocities and the observed first-person POV video. The action is the joint torques exerted by joint actuators. The transition probability is the humanoid dynamics provided by the physics simulation environment. In our imitation learning framework, the reward function is based on the similarity between the generated pose and its corresponding training pose. Based on this MDP, we perform imitation learning (IL) to obtain a humanoid control policy that is conditioned on the egocentric video. Once an optimal policy is learned, it can be used to generate a physically-grounded pose sequence given an egocentric video sequence.

The use of imitation learning to estimate pose from egocentric video requires a set of demonstrated 'expert behaviors' which in our scenario would be a set of egocentric videos labeled with 3D joint positions and joint torques. However, it is not easy to obtain such data without instrumenting the body with other sensors such as an exoskeleton [6]. Instead, we propose a two-step imitation learning process to learn a video-conditioned humanoid control policy for ego-pose estimation. In the first step, following Merel *et al.* [11], we learn a set of humanoid control policies imitating different human behaviors in motion capture data to generate virtual humanoid pose sequences, from which we can render first-person POV videos. In the second step, imitation learning is again used to learn a video-conditioned policy which maps video features to optimal joint torques, to yield a physically valid 3D pose sequence. In this way, we are able to learn a video-conditioned control policy without the need for direct measurements of joint torques from the camera wearer.

We note that the two-stage imitation learning process described thus far relies only on simulations in a virtual environment and overlooks the problem of the domain gap between the virtual and real data. Thus, we further propose to fine-tune the video-conditioned policy at test time using real data to perform domain adaptation. We use regression to estimate the best initial state that maximizes the policy's expected return and fine-tune the policy with policy gradient methods. We evaluate our approach on both virtual world data and

real-world data and show that our pose estimation technique can generalize well to real first-person POV video data despite being trained on virtual data.

In this work, we aim to show that a decision-theoretic approach to human motion estimation offers a powerful representation that can naturally map the visual input of the human visual system (*i.e.*, egocentric video) to body dynamics while taking into account the role of physics. Towards this aim, we focus on estimating the pose of human locomotion using a head-mounted camera. To the best of our knowledge, this is the first work to utilize physically grounded imitation learning to generate ego-pose estimates using a wearable camera.

2 Related work

Third-person pose estimation. Pose estimation from third-person images or videos has been studied for decades [21, 10]. Existing work leverages the fact that full human body can be seen by the third-person camera. In contrast, we consider the case where the person is entirely out of sight. Thus, existing pose estimation methods are not immediately applicable to our problem setting. Some of these methods use regression to map from images to pose parameters [1, 24, 31, 26], including the recent work DeepPose [31] that uses convolutional neural networks. It is tempting to directly apply regression-based methods to egocentric pose estimation. However, such approaches are inadequate since the egocentric images only contain information about the position and orientation of the camera. Even if the method can perfectly reconstruct the motion of the camera, the underlying human poses are still under-constrained. Without prior information as regularization, unnatural human poses will emerge. This motivates us to physically model and simulate the human, and use the human dynamics as a natural regularization.

Egocentric pose estimation. Limited amount of research has looked into the problem of inferring human poses from egocentric images or videos. Most of existing methods still assume the estimated human body or part of the body is visible [8, 9, 16, 2, 17]. The "inside-out" mocap approach of [25] gets rid of the visibility assumption and infer the 3D locations of 16 or more body-mounted cameras via structure from motion. Recently, [7] show that it is possible to estimate human pose using a single wearable camera. They construct a motion graph from the training data and recover the pose sequence by solving for the optimal pose path. In contrast, we explicitly model and simulate human dynamics, and learns a video-conditioned control policy.

Adversarial imitation learning. Our problem suits a specific setting of imitation learning in which the learner only has access to samples of expert trajectories and is not allowed to query the expert during training. Behavior cloning [15], which treats the problem as supervised learning and directly learns the mapping from state to action for each timestep, suffers from compounding error caused by covariate shift [18, 19]. Another approach, inverse reinforcement learning (IRL) [20, 12], learns a cost function by prioritizing expert trajectories over others and thus avoid the compounding error problem common in methods that fit single-

timestep decisions. However, IRL algorithms are very expensive to run because they need to solve a reinforcement learning problem in the inner loop. Generative adversarial imitation learning (GAIL; [4]) extends the GAN framework to solve this problem. A policy acts as a generator to produce sample trajectories and a discriminator is used to distinguish between expert trajectories and generated ones. It uses reinforcement learning algorithms to optimize the policy and the policy is rewarded for fooling the discriminator. The key benefit of GAIL is that no explicit hand-designed metric is needed for measuring the similarity between imitation and demonstration data.

Learning human behaviors. There have been two types of approach for modeling human movements: one is purely kinematic, and the other is physical control-based. For the former, a good amount of research models the kinematic trajectories of human from motion capture data in the absence of physics [29, 27, 5]. The latter has long been studied in graphics community and is more relevant to our scenario. Many of these methods are model-based and require significant domain expertise. With rapid development in deep reinforcement learning (Deep RL), exciting recent work has used Deep RL for locomotion of 2D creatures [13] and 3D humanoid [14]. More recently, adversarial imitation learning from motion capture data [11] has shown beautiful results. They use context variables to learn a single policy for different behaviors such as walking and running. As a follow-up work, [32] propose to learn the context variables by a variational auto encoder (VAE).

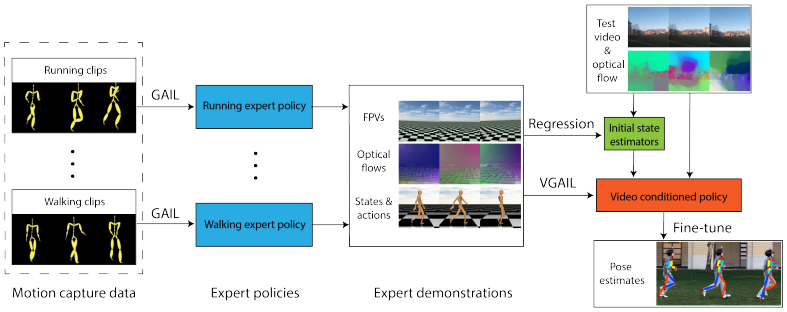

Fig. 2. Overview of our proposed pose estimation pipeline.

3 Approach

Towards our goal of estimating a physically valid 3D body pose sequence of a person using video acquired with a head-mounted camera, we propose a two-step imitation learning technique that leverages motion capture data, a humanoid model and a physics simulator. As shown in Figure 2, in our first phase, our proposed method starts with learning an initial set of C expert policies $\{\hat{\pi}_c\}_{c=1}^{C}$,

each of which represents a specific type of human behavior, *e.g.*, walking or running. In the second phase, virtual demonstrations of the humanoid are generated from each of the C policies, including state and action sequences of the humanoid, along with virtual egocentric video sequences captured by the humanoid's head-mounted camera. With this virtual expert demonstration data, we again use imitation learning to learn a video-conditioned policy that can map egocentric video features directly to joint torques which generate the pose sequence.

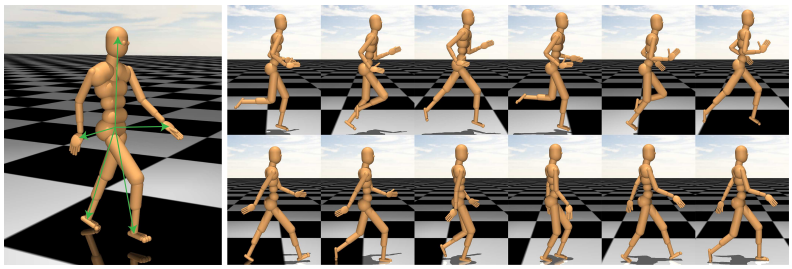

Fig. 3. Left: The humanoid model. Green arrows illustrate 3D vectors from the root to the feet, head and hands provided to the policy and discriminator. **Right:** Selected key frames of running and walking clips in motion capture data animated using the humanoid model.

Humanoid Model. By design, our underlying control policy assumes a predefined humanoid model which can be actuated in a virtual environment (see Figure 3). The humanoid model we use consists of 31 rigid bodies, 56 hinge joints and 63 degrees of freedom (DoFs). All the hinge joints can be actuated and have torque limits and range limits. The joints also have physical properties such as stiffness and damping. It is important to note here that the careful design of the humanoid is critical for solving the under-constrained problem of pose estimation from egocentric videos because the model must be similar enough to the human body for the physics simulation to match real human motion.

Humanoid Control Policy. It is common to use a Markov decision process (MDP) to model the effect of control on the dynamics of a system. In our scenario, given the humanoid model, we can formulate human(oid) motion as the output of an MDP, where any given 3D body pose sequence is assumed to be generated by an optimal policy derived from the MDP. The MDP is defined by a tuple $\mathcal{M} = \langle S, A, P, R, \gamma \rangle$, where S is the state space, A is the action (or control) space, T is the state transition dynamics, γ is the discount factor and R is the reward or cost function typically defined over the state and action space. In our formulation, the state s represents the state of the humanoid and optionally the egocentric video (second step of our learning task). The state z of the humanoid consists of the pose p and velocity v. The pose p contains the position and orientation of the root, as well as the 56 joint angles. The velocity v consists of the

linear and angular velocities of the root as well as the joint velocities. The action is composed of the joint torques of all actuated hinge joints. The dynamics of the humanoid is denoted by $P(s_{t+1}|s_t, a_t)$ (*i.e.*, how the control or action a affects the pose transition) which is determined by the simulation environment (we use the MuJoCo simulator [30]).

The solution of a given MDP is an optimal policy π that maximizes the expected return. We use $\pi(a|s)$ to denote the policy, which outputs the probability of choosing action $a \in A$ when the agent is in state $s \in S$. We use a multivariate normal distribution to model the policy π where the mean and log standard deviation are parameterized by neural networks. In our final task, we want to learn a video-conditioned policy that maps humanoid state z and egocentric video $V_{1:T}$ to joint torques, to estimate a physically valid 3D pose sequence. In what follows, we describe a two-step imitation learning method for learning this video-conditioned policy.

3.1 Stage 1: Imitation Learning for Data Generation

Instead of directly generating virtual egocentric POV videos using motion capture data, we propose to first learn a set of expert control policies imitating the human behaviors from the motion capture data and then use the expert policy for egocentric video generation. This provides two advantages. First, motion capture data is often noisy and our humanoid model cannot perfectly match the real human motion sequence. In contrast, an expert policy successfully learned from motion capture data can generate pose sequences that are noise free and realizable by our humanoid model. Second, the imitation learning procedure solves the inverse dynamics problem (*i.e.*, the control policy $\pi(a|s)$ is learned from observed state transition dynamics $p(s'|s)$) and the policy provides the joint torques for generating novel pose sequences and egocentric videos, which we show later is needed for learning the video-conditioned policy.

Our method first learns a set of expert policies $\{\hat{\pi}_c\}_{c=1}^{C}$ from motion capture data using generative adversarial imitation learning (GAIL) following Merel *et al.* [11]. Each of the expert policies represents a specific type of human behavior. In this stage, the state s of the MDP is just the state of the humanoid z as no video input is involved. Similar to GAN, the loss function of GAIL takes the form:

$$\ell(\theta, \phi) = \mathbb{E}_{z\sim\pi_\theta}[\log(1 - D_\phi(z))] + \mathbb{E}_{\hat{z}\sim\hat{\pi}}[\log(D_\phi(\hat{z}))], \tag{1}$$

where π_θ is the policy we want to learn and $\hat{\pi}$ is the expert policy implicitly represented by expert demonstrations $\{\hat{z}_i\}_{i=1}^{N}$. At each iteration, the policy acts as a generator to collect samples $\{z_i\}_{i=1}^{M}$ and rewards $\{r_i\}_{i=1}^{M}$. Using these samples and rewards, policy gradient methods (*e.g.*, TRPO [22], PPO [23]) are employed to update the policy and thus decrease the loss ℓ w.r.t θ. Once the generator update is done, we also need to update the discriminator to distinguish between generated samples and expert demonstrations. As argued by Merel *et al.* [11], using the full state z of the humanoid performs poorly because our simplified

Algorithm 1 Video-conditioned generative adversarial imitation learning

Input: Set of expert demonstrations $\{\hat{\tau}_i\}_{i=1}^N$
Output: Learned policy $\pi_\theta\left(a|z, V_{1:T}\right)$
Randomly Initialize policy π_θ and discriminator D_ϕ
repeat
  // Perform generator updates
  for k in $1 \dots N$ **do**
    Sample an expert trajectory $\hat{\tau}_k$ from $\{\hat{\tau}_i\}_{i=1}^N$
    Conditioned on $\hat{V}^k_{1:\hat{T}_k}$, execute policy π_θ to collect learner's trajectory τ_k
    Compute rewards $r_t^k = -\log\left(1 - D_\phi(z_t^k, \hat{V}^k_{1:\hat{T}_k})\right) - \alpha||p_t^k - \hat{p}_t^k||_2 + \beta$
  end for
  Update θ by policy gradient methods (e.g. TRPO, PPO) using rewards $\{r_t^k\}$
  // Perform discriminator updates
  for j in $1 \dots J$ **do**
    $\ell(\phi) = \frac{1}{N}\sum_{k=1}^N \left[\frac{1}{T_k}\sum_{t=1}^{T_k}\log\left(1 - D_\phi(z_t^k, \hat{V}^k_{1:\hat{T}_k})\right) + \frac{1}{\hat{T}_k}\sum_{t=1}^{\hat{T}_k}\log\left(D_\phi(\hat{z}_t^k, \hat{V}^k_{1:\hat{T}_k})\right)\right]$
    Update ϕ by a gradient method w.r.t. $\ell(\phi)$
  end for
until Max iteration reached

humanoid model cannot perfectly match the real human. Thus, we only use a partial state representation of the humanoid as state input z to both the policy and discriminator. Our partial state includes the root's linear and rotational velocities axis-aligned to the root orientation frame, upward direction of the root, as well as 3D displacement vectors from the root to each foot, each hand, and head, also in the root coordinate frame (see Figure 3 (Left)). We also added the orientation of the head in the root coordinate to the partial state for GAIL to learn natural head motions. After we train expert policies $\{\hat{\pi}_c\}_{c=1}^C$ using GAIL, we can generate a large amount of expert trajectories $\{\hat{\tau}_i\}_{i=1}^N$ from different human behaviors, where each expert trajectory $\hat{\tau}_i$ contains a state sequence $\hat{z}^i_{1:\hat{T}_i}$, an action sequence $\hat{a}^i_{1:\hat{T}_i}$ and a virtual egocentric video sequence $\hat{V}^i_{1:\hat{T}_i}$.

3.2 Stage 2: Imitation Learning for Ego-Pose Estimation

Using the expert trajectories $\{\hat{\tau}_i\}_{i=1}^N$ generated in the first stage, we can now learn a video-conditioned policy $\pi_\theta\left(a|z, V_{1:T}\right)$ with our video-conditioned GAIL (VGAIL) algorithm outlined in Algorithm 1. As we only care about the motion of the camera, we extract optical flow from egocentric videos and overload the notation to use optical flow as the video motion features $V_{1:T}$. In this stage, the state s of the MDP is the combination of the state z of the humanoid and the egocentric optical flow $V_{1:T}$. The VGAIL loss becomes

$$\ell(\theta, \phi) = \mathbb{E}_{z\sim\pi_\theta}\left[\log\left(1 - D_\phi(z, \hat{V}_{1:T})\right)\right] + \mathbb{E}_{\hat{z}\sim\hat{\pi}}\left[\log\left(D_\phi\left(\hat{z}, \hat{V}_{1:T}\right)\right)\right]. \quad (2)$$

We use $\hat{V}_{1:T}$ in above equation since the policy is trained on egocentric videos in expert demonstrations. In GAIL, expert demonstrations are a set of expert states $\{\hat{z}_i\}$ of the humanoid and their temporal correlation is dismissed. In VGAIL, expert demonstrations become a set of expert trajectories $\{\hat{\tau}_i\}$ with sampled expert trajectory $\hat{\tau}_k$ containing a state sequence $\hat{s}^k_{1:\hat{T}_k}$ (poses $\hat{p}^k_{1:\hat{T}_k}$ and velocities

$\hat{v}^k_{1:\hat{T}_k}$), an action sequence $\hat{a}^k_{1:\hat{T}_k}$ and a video sequence $\hat{V}^k_{1:\hat{T}_k}$. This provides two benefits. First, as we want our policy-generated pose sequence $p^k_{1:T_k}$ to match with the expert pose sequence, we use the expert pose sequence $\hat{p}^k_{1:\hat{T}_k}$ to augment the reward with an additional pose alignment term $-||p^k_t - \hat{p}^k_t||_2$, which uses L2-norm to penalize pose difference. Second, we can use the action sequence $\hat{a}^k_{1:\hat{T}_k}$ to pre-train the policy with behavior cloning [15], which accelerates the training significantly. The reward for VGAIL is

$$r^k_t = -\log\left(1 - D_\phi(z^k_t, \hat{V}^k_{1:T_k})\right) - \alpha||p^k_t - \hat{p}^k_t||_2 + \beta\,, \tag{3}$$

where α is a weighting coefficient and β is a 'living' bonus to encourage longer episode (the simulation episode will end if the humanoid falls down). α and β are set to 3.0 and 5.0 respectively in our implementation.

Again, we use the partial state of the humanoid discussed in Section 3.1 as humanoid state z to both the policy and discriminator. As shown in Figure 4(Bottom), for both the policy and discriminator networks, we use a CNN to extract visual motion features and pass them to a bidirectional LSTM to process temporal information, and a multilayer perceptron (MLP) following the LSTM outputs the action distribution (policy) or the classification probability (discriminator). Once the video-conditioned policy $\pi_\theta\,(a|z, V_{1:T})$ is learned, given an egocentric video with its optical flow $V_{1:T}$ and the initial state of the humanoid, we execute the policy π_θ inside the physics simulator and always choose mean actions to generate the corresponding pose sequence of the video.

3.3 Initial State Estimation and Domain Adaptation

The straightforward use of the video-conditioned policy on real egocentric video data will lead to failure for two reasons. First, without a mechanism for reliably estimating the initial state z_1 of the humanoid, the actions generated by the policy cause the humanoid to fall down in the physics simulator because it cannot reconcile extreme offset between the phase of the body motion and the video motion. Second, the visual features learned from the optical flow in the virtual world (checkered floor and sky box) is usually very different from the environment in real egocentric videos, and therefore the policy is not able to accurately interpret the optical flow. We propose two important techniques to enable pose estimation with real-world video data.

Initial state estimation. We propose to learn a set of state estimators $\{f_c\}_{c=1}^C$ where f_c maps an optical flow $V_{1:T}$ to its corresponding state sequence $z_{1:T}$ and is learned using expert trajectories generated by expert policy $\hat{\pi}_c$. The state at time t can be extracted by $f_c(V_{1:T})_t$. f_c is implemented as the state estimation network in Figure 4(Bottom). Visual motion features from the optical flow are extracted by a CNN and passed to a bidirectional LSTM before going into a multilayer perceptron (MLP) which makes the state predictions. We use the mean square error as loss which takes the form $l_c(\psi) = \frac{1}{T}\sum_{t=1}^{T}||f_c(V_{1:T})_t - z_t||^2$, where ψ is the parameters of f_c. We can get an optimal f_c by a SGD-based method. The

state estimators are used for initial state estimation in the policy fine-tuning step described below.

Policy fine-tuning. Our imitation learning framework allows us to fine-tune the policy on test data (of course without requiring any ground truth pose data). This fine-tuning step is essentially a reinforcement learning step that adapts the policy network to the video input $V_{1:T}$ while maximizing the reward for matching the training data distribution. In order to utilize a policy gradient method to improve and adapt the policy, we need a reward function and an initial state estimate. We define a reward function that will help to ensure that the fine-tuned policy generates pose sequences that are similar to the training data. Given the test video's optical flow $V_{1:T}$, the fine-tuning reward is defined as

$$r_t = -\log\left(1 - D_\phi(z_t, V_{1:T})\right) + \xi\,, \tag{4}$$

where ξ is a 'living' bonus (set to 0.5 in our implementation). The initial state estimate can be obtained using the state estimators described above by solving the following optimization problem:

$$c^*, b^* = \underset{c=1\ldots C,\, b=1\ldots 10}{\arg\max} \quad \mathbb{E}_{z_1=f_c(V_{1:T})_b,\, a_t\sim\pi_\theta}\left[\sum_{t=1}^{T}\gamma^t r_t\right], \tag{5}$$

where c^* is the index of the optimal estimator and b^* is the optimal start frame offset. This step enables our method to find the best initial state estimator f_{c^*} and the best start frame b^* by maximizing the expected return, where the expected return can be estimated by sampling trajectories from the video-conditioned policy. We then perform fine-tuning by sampling trajectories of the policy starting from the initial state $f_{c^*}(V_{1:T})_{b^*}$ and computing rewards using Equation 4. We employ policy gradient methods (*e.g.*, PPO [23]) to update the policy using the sampled trajectories and rewards.

4 Experimental Setup

To evaluate our proposed method's ability to estimate both accurate and physically valid pose sequence from an egocentric video, we tested our method on two datasets. The first one is a synthetic dataset using the same expert policies we learned in Section 3.1. The synthetic dataset will allow us to evaluate the accuracy of the 3D pose estimates and control actions since we have access to the ground truth through the simulator. The second dataset is composed of real-world first-person videos of different people walking and running. Our aim is to show the robustness of our technique through domain adaptation and initial pose estimation for real-world videos. Evaluations, however, are based on noisy ground truth estimates using a 2D projection of joint positions using a second static camera.

Baselines. We compare our methods against two baselines:

1. **Pose regression:** direct regression from video motion features to poses. Similar to the initial state estimation in Section 3.3, pose regression learns a

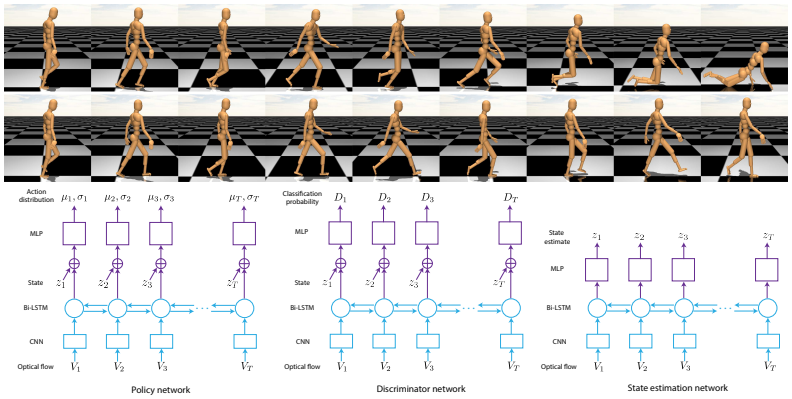

Fig. 4. Top: Humanoid falling down due to the error in initial state estimate. **Mid:** After fine-tuning for 20 iterations, the policy can generate correct walking estimates. **Bottom:** Network architecture for the policy, discriminator and state estimator. All three networks employ the same architecture for processing the optical flow: a CNN with three convolutional layers of kernel size 4 and stride 4 is used and the size of its hidden channels are (32, 32, 8), and a bidirectional LSTM is used to distill temporal information from the CNN features. For the policy and discriminator, we concatenate the LSTM output with the humanoid state z and pass it to a MLP with hidden size (300, 300, 200, 100), which outputs the action distribution (policy) or classification probability (discriminator). For the state estimation network, the LSTM output is passed to a MLP with hidden size (300, 300, 200) which outputs the state estimate.

mapping from egocentric optical flow $V_{1:T}$ to its corresponding pose sequence $p_{1:T}$. The regression network is the same as the state estimation network in Figure 4(Bottom), except the final outputs are poses instead of states.

2. **Path pose:** an adaptation of the method proposed by Jiang and Grauman [7]. This method maps a sequence of planar homographies to poses along with temporal conditional random field (CRF) smoothing to estimate the pose sequence. We do not use static scene cues as the original work since our training data is synthetic.

Both of these baselines do not impose any physical constraints on their solutions but rather attempt to directly estimate body poses.

Evaluation Metrics. To evaluate the accuracy and physical soundness of all methods, we use both pose-based and physics-based metrics:

1. **Pose error:** Pose-based metric that measures the euclidean distance between the generated pose sequence $p_{1:T}$ and the true pose sequence $\hat{p}_{1:T}$. It can be calculated as $\frac{1}{T}\sum_{t=1}^{T}||p_t - \hat{p}_t||_2$.
2. **2D projection error:** Pose-based metric used for real-world dataset where the ground-truth 3D pose sequence of the person is unknown. We project 3D joint locations of our estimated pose into a 2D image plane using a side-view virtual camera. The 2D projection error can be calculated as

$\frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}||q_t^j - \hat{q}_t^j||_2$ where q_t^j is the j-th joint's 2D position of our estimated pose and $\hat{q}_t^j$ is the ground-truth. We use OpenPose [3] to extract the ground-truth 2D joint positions from the side-view video. To comply with OpenPose, we only evaluate 12 joints (hips, knees, ankles, shoulders, elbows and wrists). For the 2D poses from our method and OpenPose, we align their positions of the center of the hips and scale the 2D coordinates to make the distance between shoulder and hip equals 0.5.

3. **Velocity error:** Physics-based metric that measures the euclidean distance between generated velocity sequence $v_{1:T}$ and true velocity sequence $\hat{v}_{1:T}$. It can be calculated as $\frac{1}{T}\sum_{t=1}^{T}||v_t - \hat{v}_t||_2$. v_t can be approximated by $(p_{t+1} - p_t)/h$ using finite difference method where h is the time step and $\hat{v}_t$ is computed in the same fashion.
4. **Smoothness:** Physics-based metric that uses average magnitude of joint accelerations to measure the smoothness of the generated pose sequence. It can be calculated as $\frac{1}{TG}\sum_{t=1}^{T}||a_t||_1$ where G is the number of actuated DoFs and a_t can be approximated by $(v_{t+1} - v_t)/h$.

4.1 Implementation Details

Motion capture data and simulation. We use CMU graphics lab motion capture database to learn expert policies as described in Section 3.1. The humanoid is similarly constructed as the CMU humanoid model in DeepMind control suite [28] with tweaks on joint stiffness, damping and torque limits. We learn 4 expert policies from 4 clips (0801, 0804, 0807, 0901) of the motion capture data corresponding to three styles of walking (slow, normal, fast) and one style of running. The physics simulation environment has a simulation timestep of 6.67ms and a control timestep of 33.3ms (control changes after 5 simulation steps).

Imitation learning parameters. The video-conditioned policy is pre-trained using behavior cloning for 100 iterations. In VGAIL, at every iteration, the policy generates sample trajectories with a total batch size of 50k timesteps. We perform online z-filtering of state inputs for normalization. The standard deviation for each action dimension is initialized to 0.1. The reward is clipped with a max value of 10 and advantages are normalized. For policy optimization, we use proximal policy optimization (PPO [23]) with a 0.2 clipping threshold. The discount factor γ is 1. The learning rate for the policy and discriminator is 5e-5 and 1e-5 respectively with the discriminator updated 5 times in the inner loop. We terminate the training after 6000 iterations to prevent over-fitting. When fine-tuning the policy, we reduce the batch size to 5k and it takes about 2s per iteration on a GTX 1080Ti.

5 Virtual World Validation

We first evaluate our method on a test dataset generated using expert policies learned in Section 3.1. The dataset consists of 20 trajectories, each of which is 100 timesteps long. The policy is fine-tuned for 20 iterations for each sequence.

	Smoothness	Velocity error	Pose error
Ours	**11.9876**	**6.5143**	0.9779
Pose regression	36.1628	9.0611	**0.8310**
Path pose [7]	198.6509	45.0189	1.7643
	Smoothness	Velocity error	Pose error
Ours	**11.9876**	6.5143	0.9779
Ours-IE	12.2472	7.5337	1.2219
Ours-GTI	12.2968	**6.0761**	**0.6688**
	Smoothness	2D projection error	
Ours	**11.54**	**0.1325**	
Pose regression	44.11	0.1621	
Path pose [7]	214.21	0.1738	

Table 1. Top: Results for pose-based and physics-based metrics on virtual test dataset. **Mid:** Ablation Study. (Ours-GTI) our method with ground-truth initial state. (Ours-IE) Our method with estimated initial state before fine-tuning. **Bottom:** Results for physics-based and pose-based metrics on real-world data.

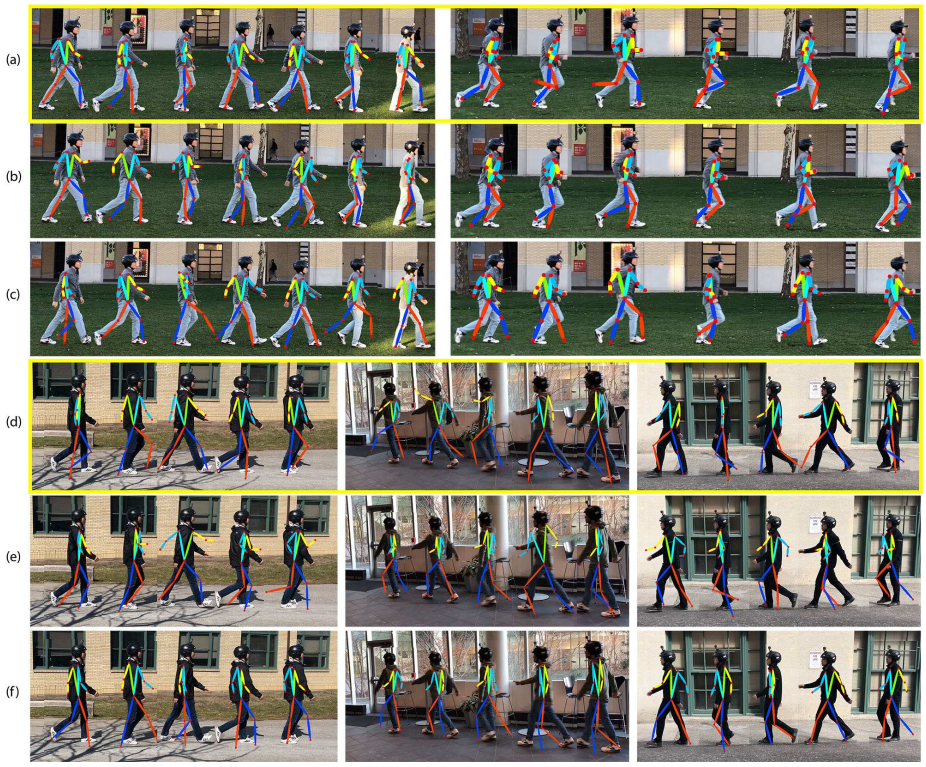

Fig. 5. Qualitative results on real world dataset. (a)(d) Our method (yellow box); (b)(e) Pose regression; (c)(f) Path pose [7]. Yellow and orange bones correspond to the left arm and leg respectively.

Table 1(Top) shows a comparison of our method against the two baselines (pose regression and path pose). We observe that our method outperforms the baselines in terms of physics-based metrics (acceleration and velocity error), and the pose estimation is reasonably accurate.

Ablation Study. As shown in Table 1(Mid), the accuracy of initial state plays an important role in our method. As expected, our method with ground-truth initial state is much more accurate than with estimated initial state. This is because sometimes the humanoid falls down due to the error in the initial state estimate as shown in Figure 4(Top). Our fine-tuning approach can adapt the policy to recover from the error in initial state and generate a more accurate pose sequence (see Figure 4(Mid)).

6 Real World Validation

To understand the true utility of our approach, we must evaluate its performance on real-world first-person videos. In this experiment, we apply our virtually trained video-conditioned policy on real video data and show that our approach is able to estimate both accurate and physically-valid pose sequences. Since we do not have access to the true 3D poses of the person recording the egocentric video, we use a secondary static camera (third-person POV) to measure the error of our pose estimation based on 2D projections of joint positions.

We evaluate our proposed method on 12 video sequences composed of 3 different people performing the walking activity and running activity, in both outdoor and indoor scenes. Each egocentric video is 3-7 seconds long and obtained by a head-mounted GoPro camera. For each sequence, the policy is fine-tuned for 50 iterations. As indicated in Table 1(Bottom), our method estimates much smoother (3.8x, 18.5x) pose sequences and is also more accurate in terms of 2D projection error (18%, 24%). Figure 5 shows a qualitative comparison of our approach against the two baselines.

7 Conclusion

We proposed a physically-grounded approach for ego-pose estimation that learns a video-conditioned control policy to generate the pose estimates in physics simulation. We evaluated our method on both simulation data and real-world data and show that our approach significantly outperforms baseline methods in terms of physics-based metrics and is also accurate. Our experiments also demonstrated the effectiveness of our proposed fine-tuning approach for domain adaptation from synthetic to real data. We believe our work is one of the first to open new research directions that consider the role of physics in understanding human motion in computer vision.

Acknowledgment. This work was sponsored in part by JST CREST (JPMJCR14E1) and IARPA (D17PC00340).

References

1. Agarwal, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 2, pp. II–II. IEEE (2004)
2. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. In: ACM Transactions on Graphics (TOG). vol. 22, pp. 402–408. ACM (2003)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. vol. 1, p. 7 (2017)
4. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: Advances in Neural Information Processing Systems. pp. 4565–4573 (2016)
5. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG) **36**(4), 42 (2017)
6. Hwang, B., Jeon, D.: A method to accurately estimate the muscular torques of human wearing exoskeletons by torque sensors. Sensors **15**(4), 8337–8357 (2015)
7. Jiang, H., Grauman, K.: Seeing invisible poses: Estimating 3d body pose from egocentric video. arXiv preprint arXiv:1603.07763 (2016)
8. Li, C., Kitani, K.M.: Model recommendation with virtual probes for egocentric hand detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2624–2631 (2013)
9. Li, C., Kitani, K.M.: Pixel-level hand detection in ego-centric videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3570–3577 (2013)
10. Liu, Z., Zhu, J., Bu, J., Chen, C.: A survey of human pose estimation: the body parts parsing based methods. Journal of Visual Communication and Image Representation **32**, 10–19 (2015)
11. Merel, J., Tassa, Y., Srinivasan, S., Lemmon, J., Wang, Z., Wayne, G., Heess, N.: Learning human behaviors from motion capture by adversarial imitation. arXiv preprint arXiv:1707.02201 (2017)
12. Ng, A.Y., Russell, S.J., et al.: Algorithms for inverse reinforcement learning. In: Icml. pp. 663–670 (2000)
13. Peng, X.B., Berseth, G., Van de Panne, M.: Terrain-adaptive locomotion skills using deep reinforcement learning. ACM Transactions on Graphics (TOG) **35**(4), 81 (2016)
14. Peng, X.B., Berseth, G., Yin, K., Van De Panne, M.: Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. ACM Transactions on Graphics (TOG) **36**(4), 41 (2017)
15. Pomerleau, D.A.: Efficient training of artificial neural networks for autonomous navigation. Neural Computation **3**(1), 88–97 (1991)
16. Ren, X., Gu, C.: Figure-ground segmentation improves handled object recognition in egocentric video. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3137–3144. IEEE (2010)
17. Rogez, G., Supancic, J.S., Ramanan, D.: First-person pose recognition using egocentric workspaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4325–4333 (2015)
18. Ross, S., Bagnell, D.: Efficient reductions for imitation learning. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 661–668 (2010)
19. Ross, S., Gordon, G.J., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: International Conference on Artificial Intelligence and Statistics. pp. 627–635 (2011)

20. Russell, S.: Learning agents for uncertain environments. In: Proceedings of the eleventh annual conference on Computational learning theory. pp. 101–103. ACM (1998)
21. Sarafianos, N., Boteanu, B., Ionescu, B., Kakadiaris, I.A.: 3d human pose estimation: A review of the literature and analysis of covariates. Computer Vision and Image Understanding **152**, 1–20 (2016)
22. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: Proceedings of the 32nd International Conference on Machine Learning (ICML-15). pp. 1889–1897 (2015)
23. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
24. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: null. p. 750. IEEE (2003)
25. Shiratori, T., Park, H.S., Sigal, L., Sheikh, Y., Hodgins, J.K.: Motion capture from body-mounted cameras. In: ACM Transactions on Graphics (TOG). vol. 30, p. 31. ACM (2011)
26. Sminchisescu, C., Kanaujia, A., Metaxas, D.N.: Bm^3e: Discriminative density propagation for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(11) (2007)
27. Sussillo, D., Abbott, L.F.: Generating coherent patterns of activity from chaotic neural networks. Neuron **63**(4), 544–557 (2009)
28. Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D.d.L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al.: Deepmind control suite. arXiv preprint arXiv:1801.00690 (2018)
29. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Modeling human motion using binary latent variables. In: Advances in neural information processing systems. pp. 1345–1352 (2007)
30. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 5026–5033. IEEE (2012)
31. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1653–1660 (2014)
32. Wang, Z., Merel, J., Reed, S., Wayne, G., de Freitas, N., Heess, N.: Robust imitation of diverse behaviors. arXiv preprint arXiv:1707.02747 (2017)