Physics-based 3D Human Pose Estimation from Monocular Video

Yuxuan Mu

PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time, SIGGRAPH Asia 2020

SimPoE: Simulated Character Control for 3D Human Pose Estimation, ICCV 2021

Differentiable Dynamics for Articulated 3d Human Motion Reconstruction, CVPR 2022

3D Human Pose Estimation from Monocular Video

Background:

Progress on single-image 3D pose and shape estimation (w/ sufficient 3D annotations)

Challenge:

Inaccurate and unnatural motion sequences on video

E.g. Unreal motion, Jitter, Penetration

-> **Physical Awareness**

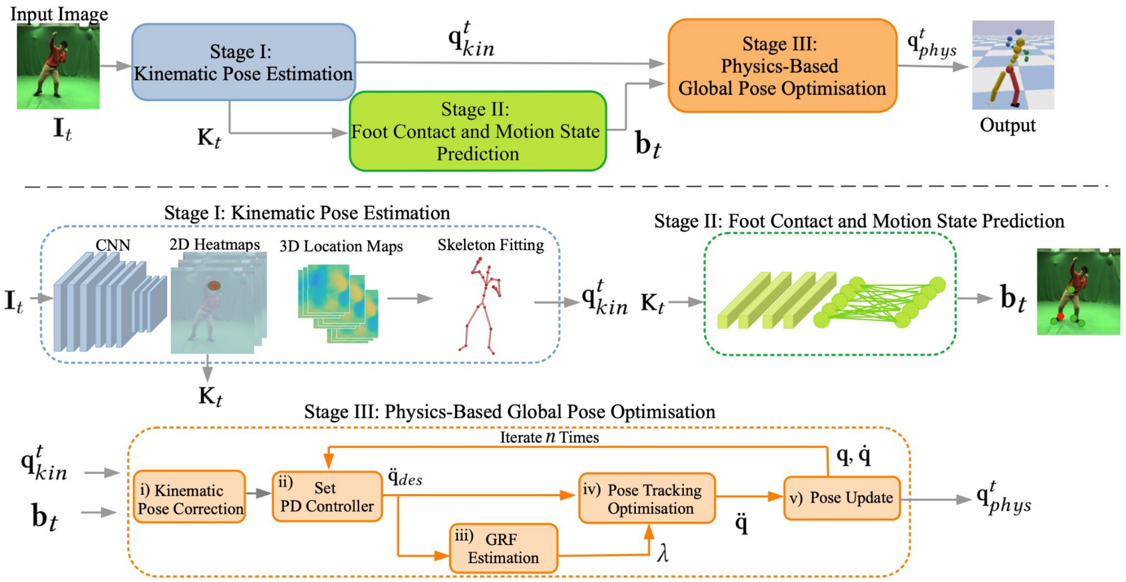
An example from VIBE*

PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time, SIGGRAPH Asia 2020

Contributions:

- The first algorithm for physically plausible, real-time and marker-less human 3D motion capture
- A CNN to detect foot contact and motion states from images
- Pose optimization framework with a human parameterised by a torque-controlled simulated character

Approach



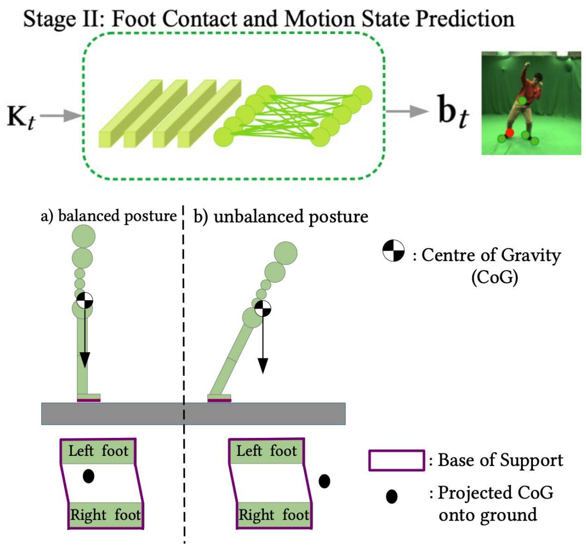
Input Image
$\mathbf{I}_t$
Stage I:
Kinematic Pose Estimation
$\mathbf{q}^t_{kin}$
$\mathbf{K}_t$
Stage II:
Foot Contact and Motion State Prediction
$\mathbf{b}_t$
Stage III:
Physics-Based
Global Pose Optimisation
$\mathbf{q}^t_{phys}$
Output
Stage I: Kinematic Pose Estimation
CNN
2D Heatmaps
3D Location Maps
Skeleton Fitting
Stage II: Foot Contact and Motion State Prediction
Stage III: Physics-Based Global Pose Optimisation
Iterate n Times
$\mathbf{q}, \dot{\mathbf{q}}$
i) Kinematic Pose Correction
ii) Set PD Controller
$\ddot{\mathbf{q}}_{des}$
iii) GRF Estimation
iv) Pose Tracking Optimisation
v) Pose Update
$\ddot{\mathbf{q}}$
λ

Approach

Stage II: Foot Contact and Motion State Detection

Foot Contact -> simulator

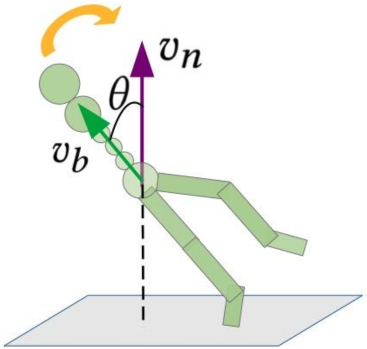
Motion State (stationary or not) -> Stage III(i) Pose Correction

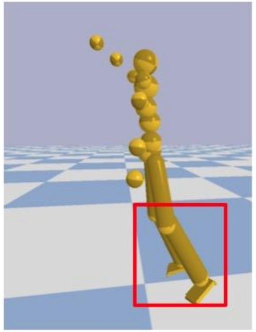


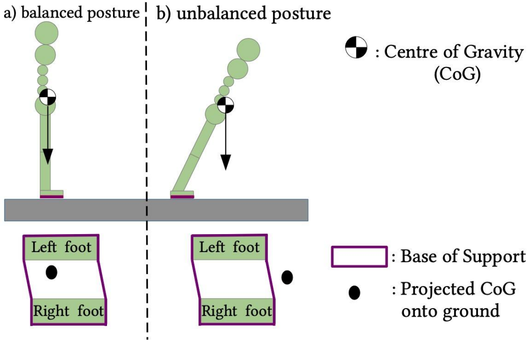
Approach

Stage III(i) Pose Correction

Performs until 1) the pose becomes non-stationary or 2) CoG projects inside BoS







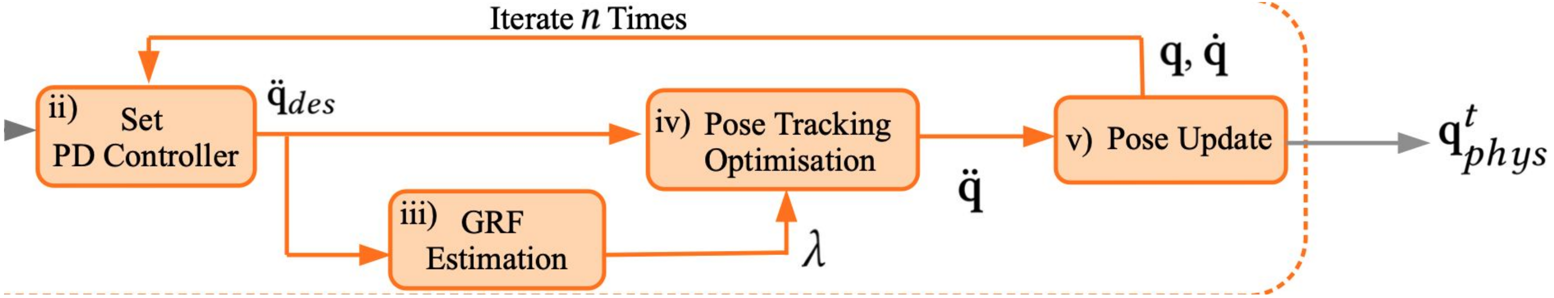
Approach

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T\mathbf{G}\lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$$ Physical Prior

Stage III Physics-Based Global Pose Optimisation

- Acceleration $\ddot{\mathbf{q}}_{des} = \ddot{\mathbf{q}}^t_{kin} + k_p(\mathbf{q}^t_{kin} - \mathbf{q}) + k_d(\dot{\mathbf{q}}^t_{kin} - \dot{\mathbf{q}})$
- Ground Reaction Force (GRF) Estimation

$$\min_{\lambda} \|\mathbf{M}_1\ddot{\mathbf{q}}_{des} + \mathbf{c}_1(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}_1^T\mathbf{G}\lambda\|,$$



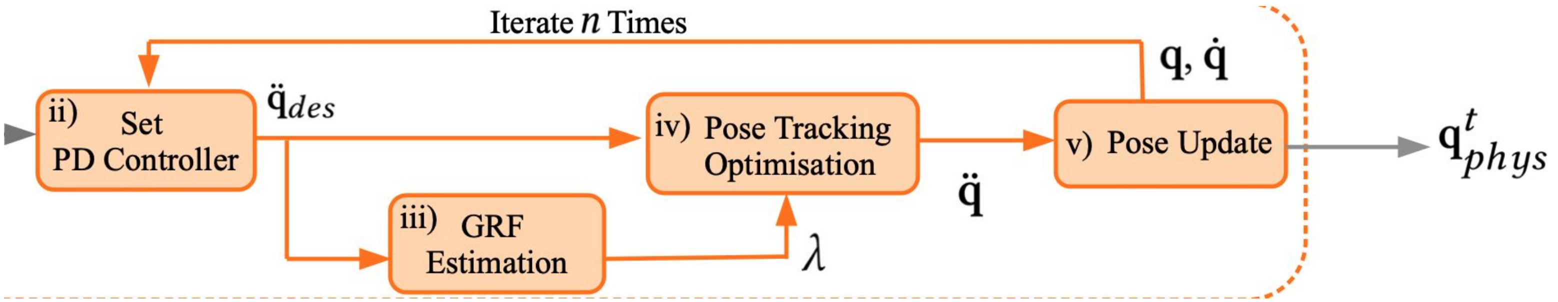
Approach

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T\mathbf{G}\lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$$ Physical Prior

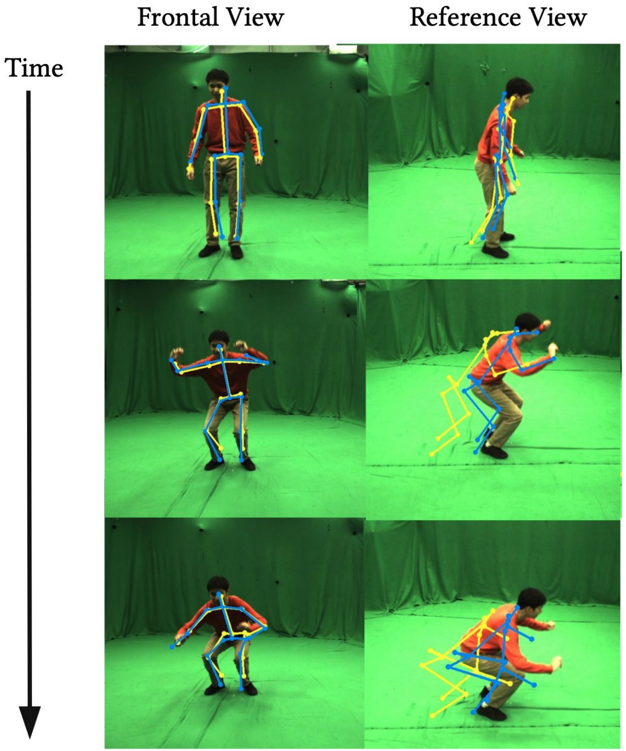
Stage III Physics-Based Global Pose Optimisation

- Physics-Based Pose Optimisation

$$\min_{\ddot{\mathbf{q}},\boldsymbol{\tau}} \|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{des}\| + \|\boldsymbol{\tau}\|,$$
$$\text{s.t. } \mathbf{M}\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T\mathbf{G}\lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \quad 0 \le \dot{r}_j^n, \quad |\dot{r}_j^t| \le \sigma, \quad \text{and} \quad |\dot{r}_j^b| \le \sigma, \quad \mathbf{J}_j\dot{\mathbf{q}} = \dot{\mathbf{r}}_j.$$



Results



SimPoE: Simulated Character Control for 3D Human Pose Estimation, ICCV 2021

Motivation:

- Physical artifacts generated by kinematic-based (body motion without physical forces) pose estimation methods
- Current physical-based methods:
  - high latency, computationally intensive
  - differentiable simulator -> need to be simple -> approximation errors
  - separate stage without learning targets
- A joint learning framework that tightly integrates image-based kinematic inference and physics-based dynamics modeling

Approach

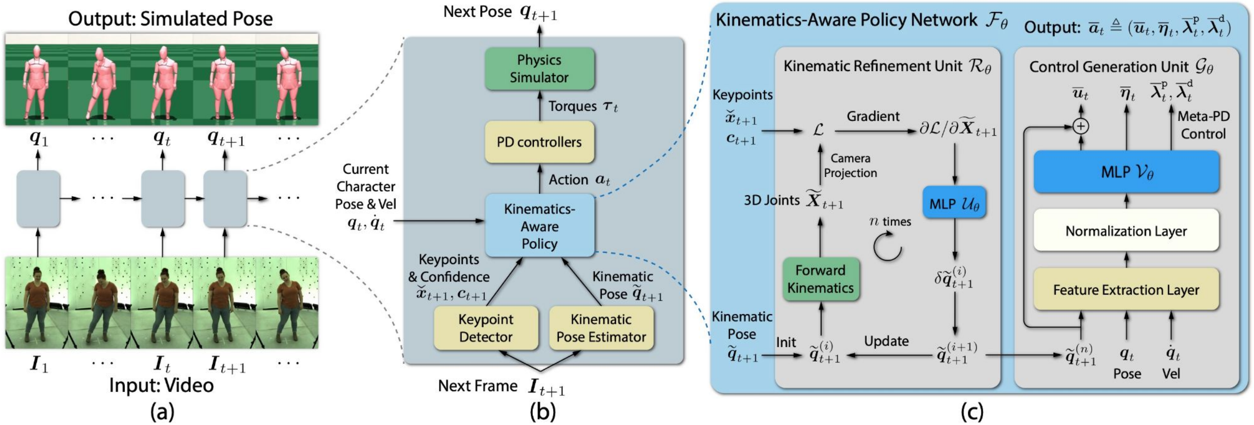
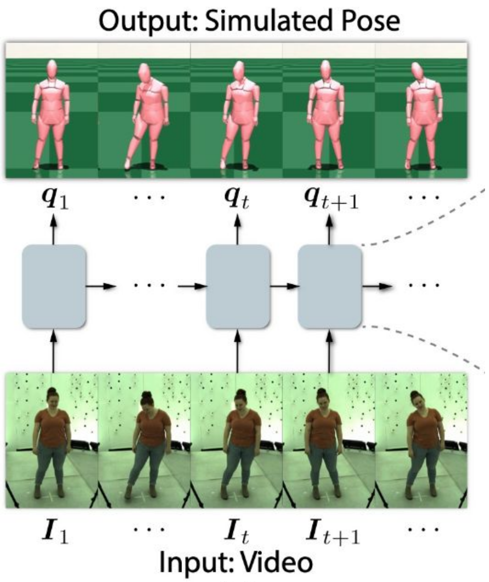


Figure 2. **Overview of our SimPoE framework.** (a) SimPoE is a physics-based causal temporal model. (b) At each frame (30Hz), the policy network $\mathcal{F}_\theta$ use the current pose $\boldsymbol{q}_t$, velocities $\dot{\boldsymbol{q}}_t$, and the next frame's estimated kinematic pose $\widetilde{\boldsymbol{q}}_{t+1}$ and keypoints $(\breve{\boldsymbol{x}}_{t+1}, \boldsymbol{c}_{t+1})$ to generate an action $\boldsymbol{a}_t$, which controls the character in the physics simulator (450Hz) via PD controllers to produce the next pose $\boldsymbol{q}_{t+1}$. (c) The policy network $\mathcal{F}_\theta$ outputs the mean action $\overline{\boldsymbol{a}}_t \triangleq (\overline{\boldsymbol{u}}_t, \overline{\boldsymbol{\eta}}_t, \overline{\boldsymbol{\lambda}}_t^{\mathrm{p}}, \overline{\boldsymbol{\lambda}}_t^{\mathrm{d}})$. The kinematic refinement unit iteratively refines a kinematic pose estimate by learning pose updates. The refined pose $\widetilde{\boldsymbol{q}}_{t+1}^{(n)}$ is used by the control generation unit to produce the mean action $\overline{\boldsymbol{a}}_t$.

Approach

1. Create a character from SMPL in the simulator
   a. Using SMPL weights to separate body parts
   b. Convex hull & constant density assumption -> Body Parts Geometry
   c. Pose: rotations



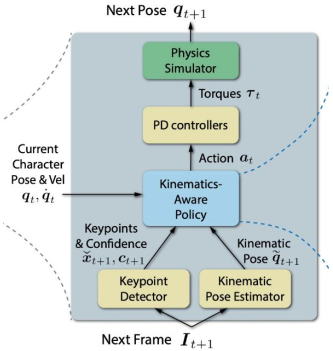
Approach

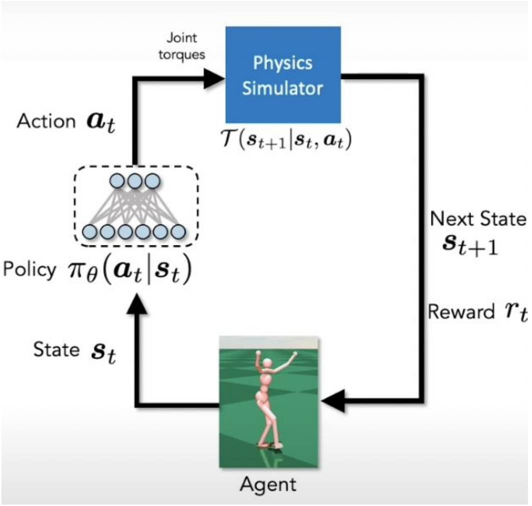
2. Simulated Character Control (RL policy solver: PPO 2017)

Definition:

States $$\boldsymbol{s}_t \triangleq (\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \widetilde{\boldsymbol{q}}_{t+1}, \check{\boldsymbol{x}}_{t+1}, \boldsymbol{c}_{t+1})$$

$\boldsymbol{q}_t$	Current pose
$\dot{\boldsymbol{q}}_t$	Joint velocities
$\widetilde{\boldsymbol{q}}_{t+1}$	Initial kinematic pose
$\check{\boldsymbol{x}}_{t+1}, \boldsymbol{c}_{t+1}$	Keypoints & conf





Approach

2. Simulated Character Control (RL policy solver: PPO 2017)

Definition: **Policy & Actions**

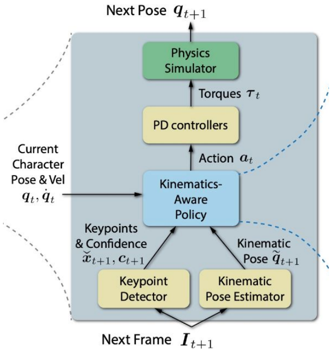
Commonly, the action is torque $\boldsymbol{\tau}_t$ o be applied to the each joint (non-root)

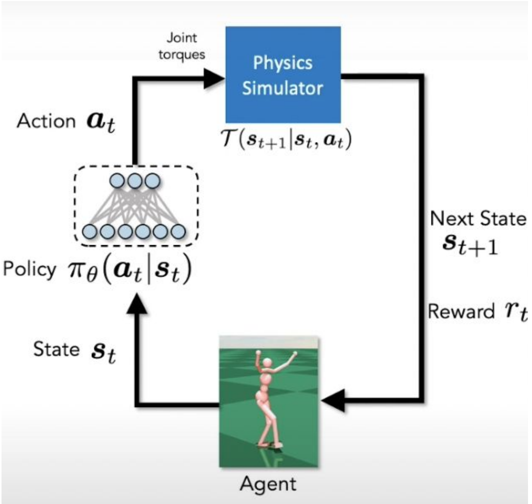
Using Proportional derivative (PD) controllers:

$$\boldsymbol{\tau}_t = \boldsymbol{k}_{\mathrm{p}} \circ (\boldsymbol{u}_t - \boldsymbol{q}_t^{\mathrm{nr}}) - \boldsymbol{k}_{\mathrm{d}} \circ \dot{\boldsymbol{q}}_t^{\mathrm{nr}}$$

$$\boldsymbol{k}_{\mathrm{p}} = \lambda_{tj}^{\mathrm{p}} \boldsymbol{k}'_{\mathrm{p}}, \quad \boldsymbol{k}_{\mathrm{d}} = \lambda_{tj}^{\mathrm{d}} \boldsymbol{k}'_{\mathrm{d}}$$

$$\boldsymbol{a}_t \triangleq (\boldsymbol{u}_t, \boldsymbol{\eta}_t, \boldsymbol{\lambda}_t^{\mathrm{p}}, \boldsymbol{\lambda}_t^{\mathrm{d}})$$





Approach

2. Simulated Character Control (RL policy solver: PPO 2017)

Definition: **Reward**

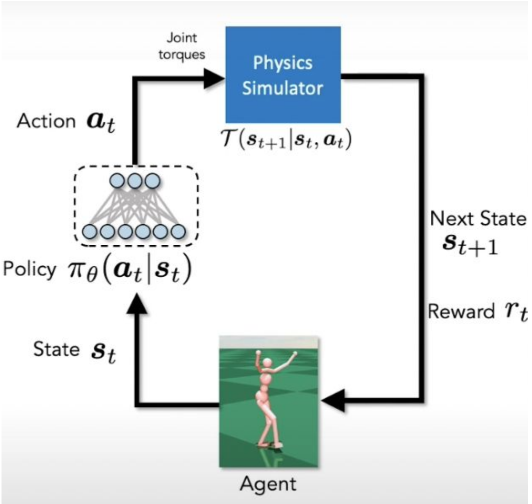
$$r_t = r_t^{\mathrm{p}} \cdot r_t^{\mathrm{v}} \cdot r_t^{\mathrm{j}} \cdot r_t^{\mathrm{k}}$$

$$r_t^{\mathrm{p}} = \exp\left[-\alpha_{\mathrm{p}}\left(\sum_{j=1}^{J}\|\boldsymbol{o}_t^j \ominus \widehat{\boldsymbol{o}}_t^j\|^2\right)\right]$$

$$r_t^{\mathrm{v}} = \exp\left[-\alpha_{\mathrm{v}}\|\dot{\boldsymbol{q}}_t - \widehat{\dot{\boldsymbol{q}}}_t\|^2\right]$$

$$r_t^{\mathrm{j}} = \exp\left[-\alpha_{\mathrm{j}}\left(\sum_{j=1}^{J}\|\boldsymbol{X}_t^j - \widehat{\boldsymbol{X}}_t^j\|^2\right)\right]$$

$$r_t^{\mathrm{k}} = \exp\left[-\alpha_{\mathrm{k}}\left(\sum_{j=1}^{J}\|\boldsymbol{x}_t^j - \widehat{\boldsymbol{x}}_t^j\|^2\right)\right]$$



Approach

3. Kinematics-Aware Policy

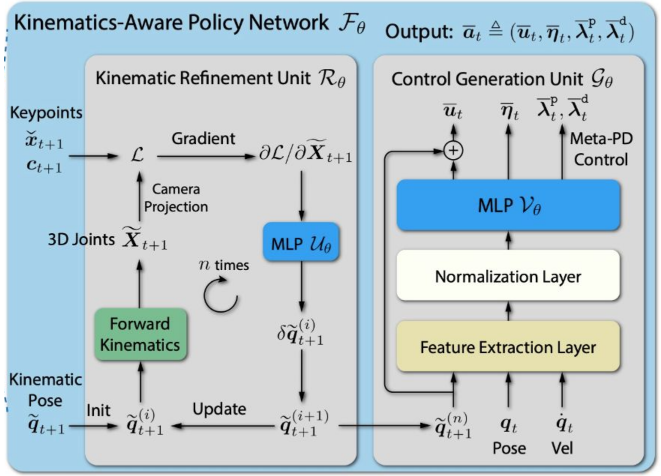
$$\text{Gaussian policy } \pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) = \mathcal{N}(\overline{\boldsymbol{a}}_t, \boldsymbol{\Sigma})$$

$$\overline{\boldsymbol{a}}_t = \mathcal{F}_\theta\left(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \widetilde{\boldsymbol{q}}_{t+1}, \breve{\boldsymbol{x}}_{t+1}, \boldsymbol{c}_{t+1}\right)$$

$$\widetilde{\boldsymbol{q}}_{t+1}^{(n)} = \mathcal{R}_\theta\left(\widetilde{\boldsymbol{q}}_{t+1}, \breve{\boldsymbol{x}}_{t+1}, \boldsymbol{c}_{t+1}\right),$$

$$(\delta\overline{\boldsymbol{u}}_t, \overline{\boldsymbol{\eta}}_t, \overline{\boldsymbol{\lambda}}_t^{\mathrm{p}}, \overline{\boldsymbol{\lambda}}_t^{\mathrm{d}}) = \mathcal{G}_\theta\left(\widetilde{\boldsymbol{q}}_{t+1}^{(n)}, \boldsymbol{q}_t, \dot{\boldsymbol{q}}_t\right),$$

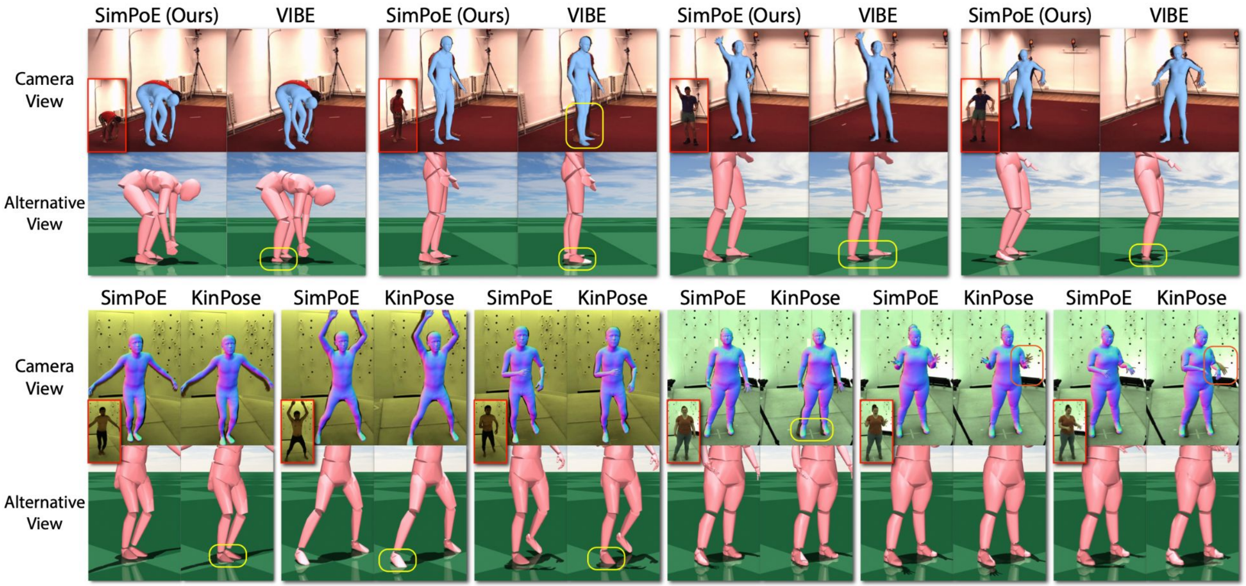
$$\overline{\boldsymbol{u}}_t = \widetilde{\boldsymbol{q}}_{t+1}^{(n)} + \delta\overline{\boldsymbol{u}}_t\,.$$



Results

Human3.6M						
Method	Physics	MPJPE ↓	PA-MPJPE ↓	Accel ↓	FS ↓	GP ↓
VIBE [21]	✗	61.3	43.1	15.2	15.1	12.6
NeurGD* [51]	✗	57.3	42.2	14.2	16.7	24.4
PhysCap [50]	✓	113.0	68.9	-	-	-
EgoPose [65]	✓	130.3	79.2	31.3	5.9	3.5
SimPoE (Ours)	✓	**56.7**	**41.6**	**6.7**	**3.4**	**1.6**
In-House Motion Dataset						
Method	Physics	MPJPE ↓	PA-MPJPE ↓	Accel ↓	FS ↓	GP ↓
KinPose	✗	49.7	40.4	12.8	6.4	3.9
NeurGD* [51]	✗	36.7	30.9	16.2	7.7	3.6
EgoPose [65]	✓	202.2	131.4	32.6	2.2	0.5
SimPoE (Ours)	✓	**26.6**	**21.2**	**8.4**	**0.5**	**0.1**

Results



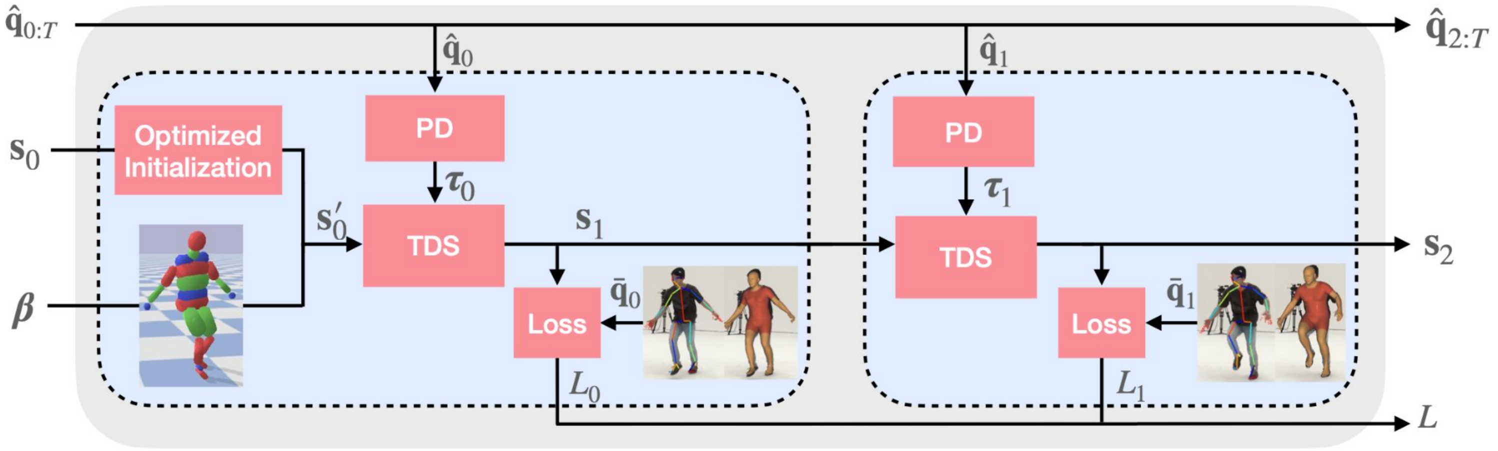
SimPoE (Ours)
VIBE
SimPoE (Ours)
VIBE
SimPoE (Ours)
VIBE
SimPoE (Ours)
VIBE
Camera View
Alternative View
SimPoE
KinPose
SimPoE
KinPose
SimPoE
KinPose
SimPoE
KinPose
SimPoE
KinPose
SimPoE
KinPose
Camera View
Alternative View

Limitation

Depends on 3D scene modeling that hinders its evaluation on in-the-wild datasets.

Its physical awareness mainly tackle the interaction between human and scene.

Differentiable Dynamics for Articulated 3d Human Motion Reconstruction, CVPR 2022



Differentiable Dynamics for Articulated 3d Human Motion Reconstruction, CVPR 2022

Method	Body	Cont.	DP	Trained	T_g	No RF
Rempe *et al.* [39]	Fixed	Feet	✗	Contacts	✗	✓
PhysCap [42]	Fixed	Feet	✓	Contacts	✓	✗
SimPoE [59]	Adapt	Full	✗	Yes	✗	✗
Shimada *et al.* [41]	Fixed	Feet	✓	Yes	✓	✗
Xie *et al.* [55]	Fixed	Feet	✓	No	✗	✓
Dynamics [15]	Adapt	Full	✗	Prior	✓	✓
DiffPhy	Adapt	Full	✓	No	✓	✓

Results

Dataset	Model	MPJPE-G	MPJPE	MPJPE-PA	MPJPE-2d	TV	Foot skate (%)
Human3.6M	VIBE [24]	207.7	68.6	43.6	16.4	0.32	27.4
	PhysCap [42]	-	97.4	65.1	-	-	-
	SimPoE [59]	-	**56.7**	**41.6**	-	-	-
	Shimada *et al.* [41]	-	76.5	58.2	-	-	-
	Xie *et al.* [55]	-	68.1	-	-	-	-
	Kinematics	145.3	83.0	55.4	13.4	0.34	47.5
	DiffPhy	**139.1**	81.7	55.6	**13.1**	**0.20**	**7.4**
AIST	Kinematics	155.7	107.4	66.9	**10.4**	0.52	50.9
	DiffPhy	**150.2**	**105.5**	**66.0**	12.1	**0.44**	**19.6**

Summary

Modeling scene interaction

Advanced simulators

Advanced learning strategies

Create a Digital Twin

Thanks