Real-time Human Motion Capture from Sparse Inertial Sensors

Xinyu Yi

Tsinghua University

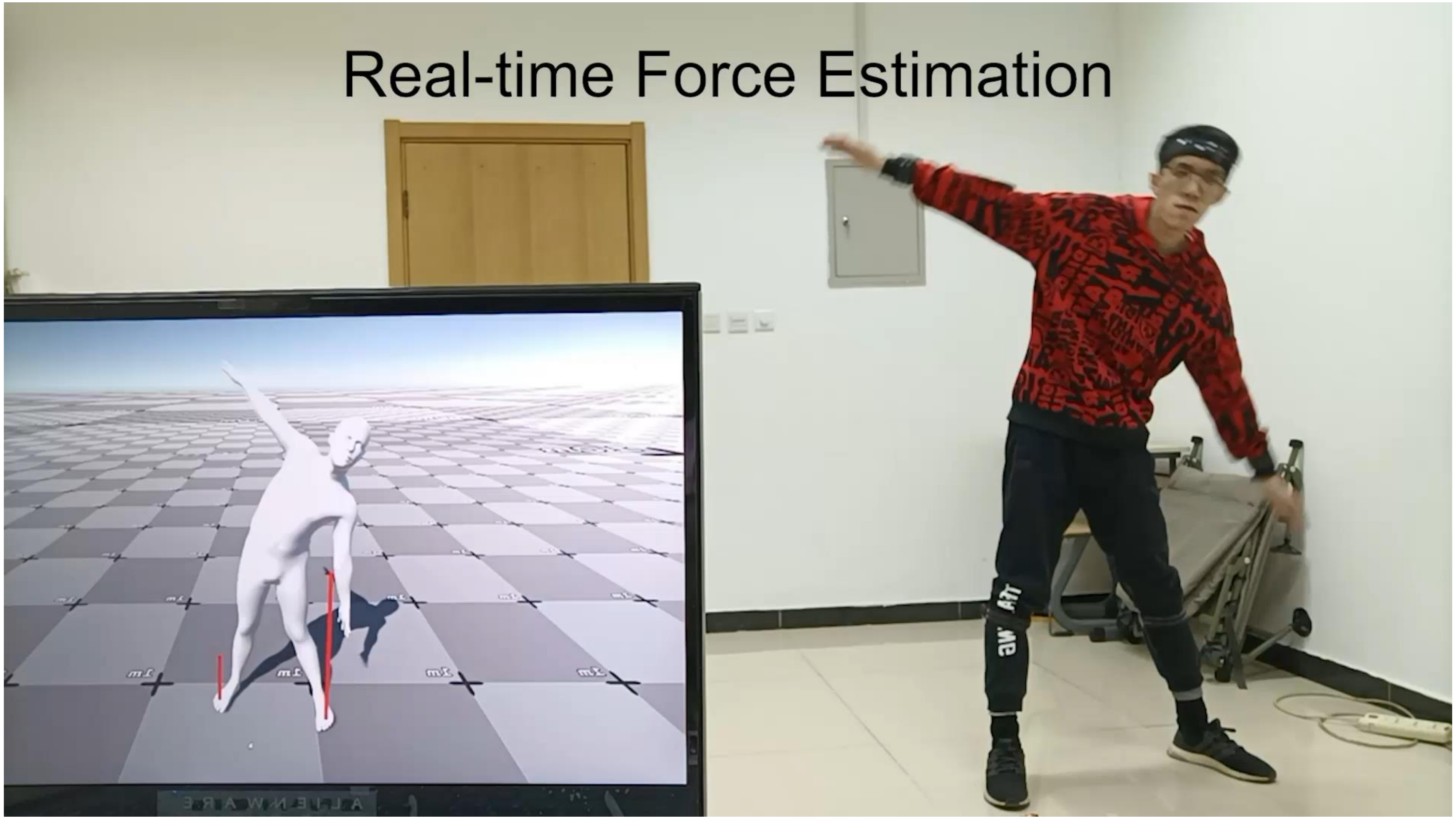
Live Demo

Our system captures real-time human pose and translation from 6 inertial sensors

Live Demo

Our system captures physically correct human motion from 6 inertial sensors

Live Demo



Our system captures joint torques and ground reaction forces from 6 inertial sensors

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)

- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)
- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)
- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)
- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)

- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Our Researches

- **TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors (SIGGRAPH 2021)**
  - Multi-stage body pose estimation (IMU -> Joints -> Pose)
  - Fusion-based global translation estimation (physics rules + neural networks)

- **PIP: Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors (CVPR 2022 Best Paper Finalist)**
  - Physics-based motion optimization
  - Learning-based RNN hidden state initialization
  - Dual PD controller: global motion control

Outline

- Introduction
- Method
- Results

INTRODUCTION

Background

- Applications of motion capture
  - Movie production
  - Augmented/Virtual reality
  - Human-computer interaction
  - Gaming
  - Sports
  - …

Background

Commercial solutions

Optical motion capture

Vicon (https://www.vicon.com/)

Inertial motion capture

Xsens (https://www.xsens.com/)

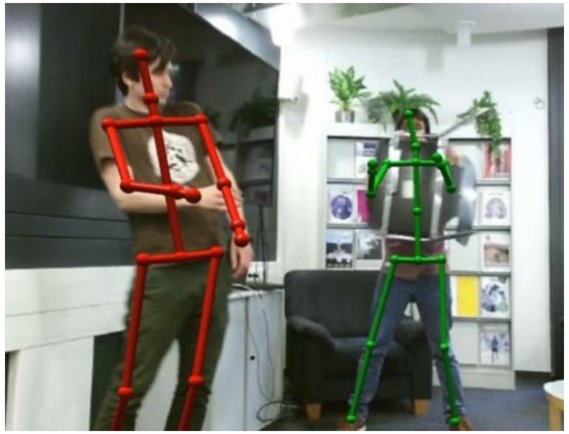
Background

Previous works

Marker-free video-based

Neural PhysCap
[Shimada et al. 2021]

Monocular Real-time Full Body Capture [Zhou et al. 2021]

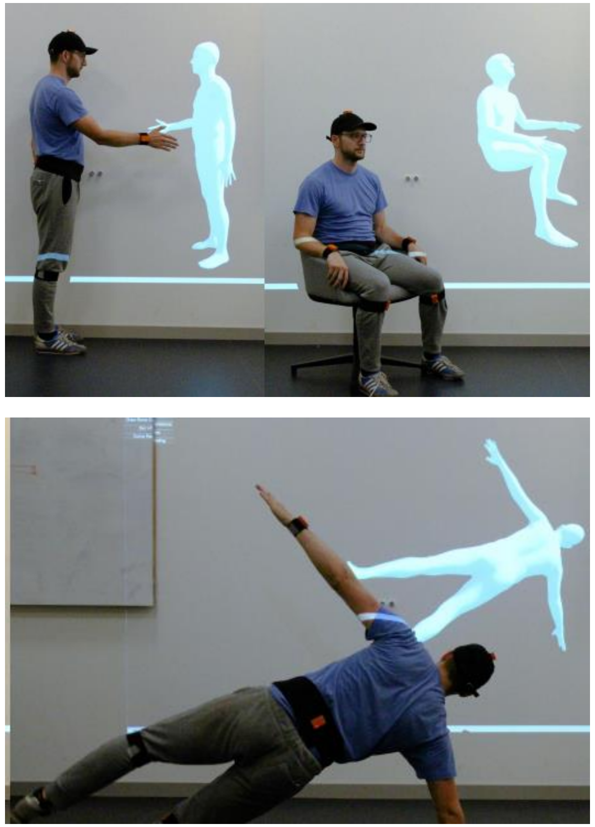


Xnect
[Mehta et al. 2020]

DeepCap
[Habermann et al. 2020]

Sparse inertial sensor-based

Sparse Inertial Poser
[Marcard et al, 2017]

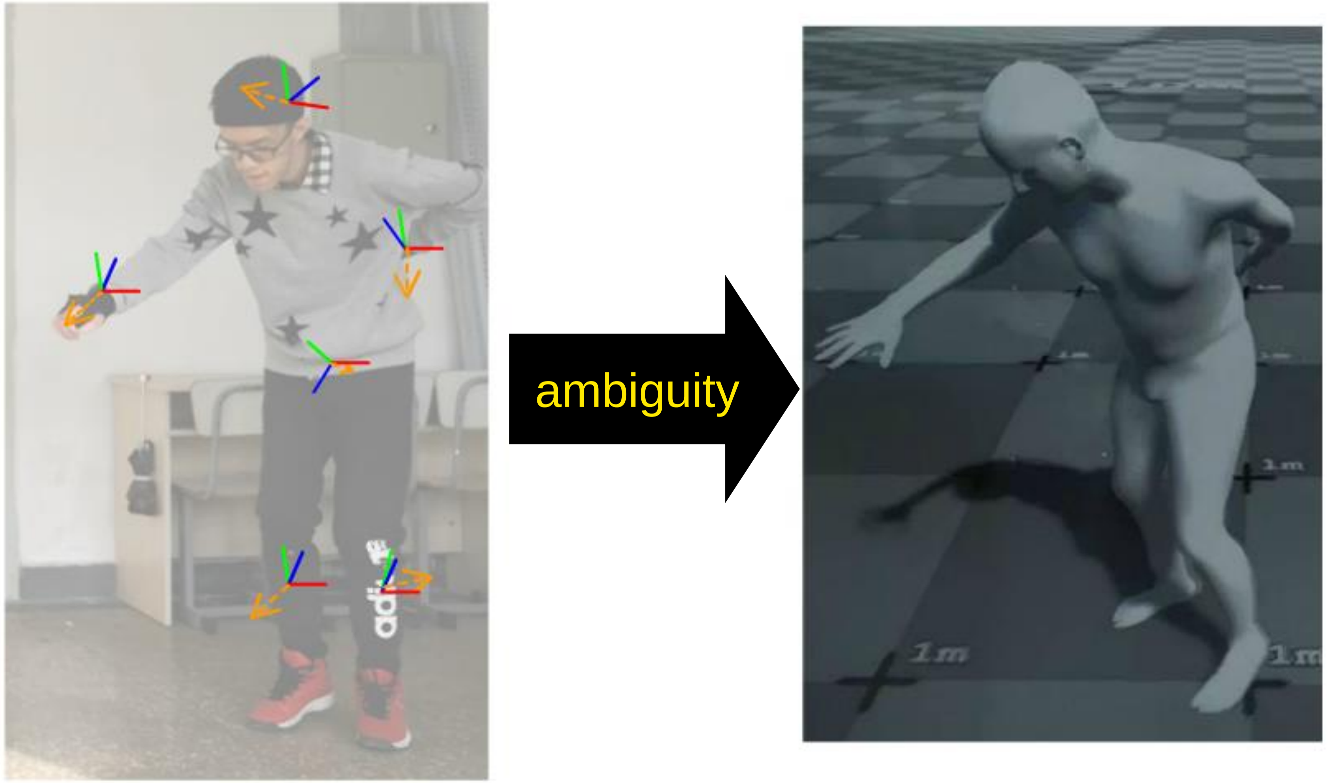


Deep Inertial Poser
[Huang et al, 2018]

Challenges

Challenges in sparse inertial mocap

- Learning pose prior
    - IMU signals are sparse and noisy
- Estimating global movements
    - No direct distance measurement
    - Acceleration signals are noisy
- Ensuring physical plausibility

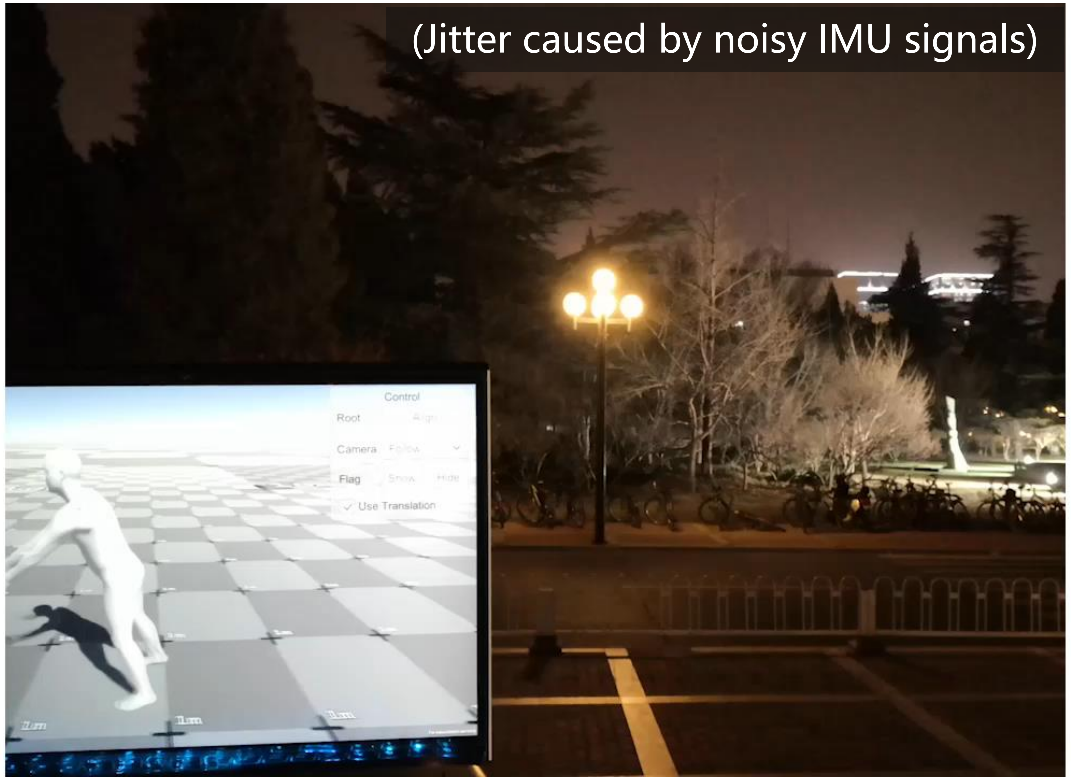


Challenges

Challenges in sparse inertial mocap

Pose ambiguity	Physical correctness

Previous works cannot disambiguate poses with similar sensor readings well

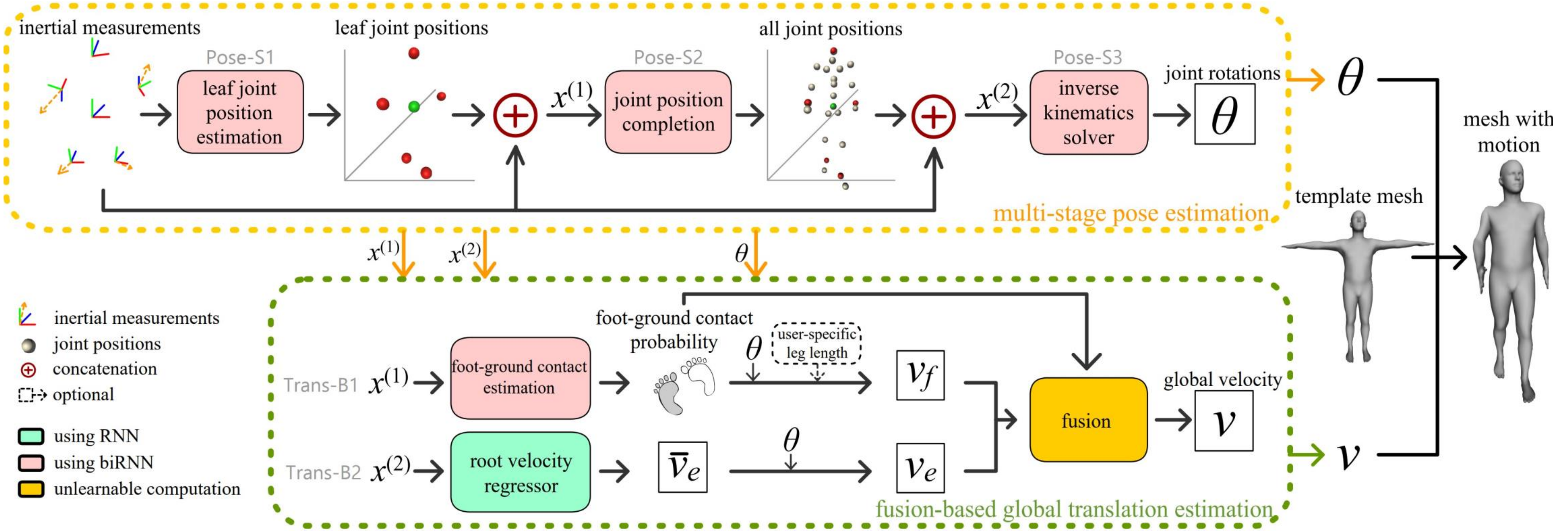


Previous works cannot ensure physical correctness of the motion

METHOD

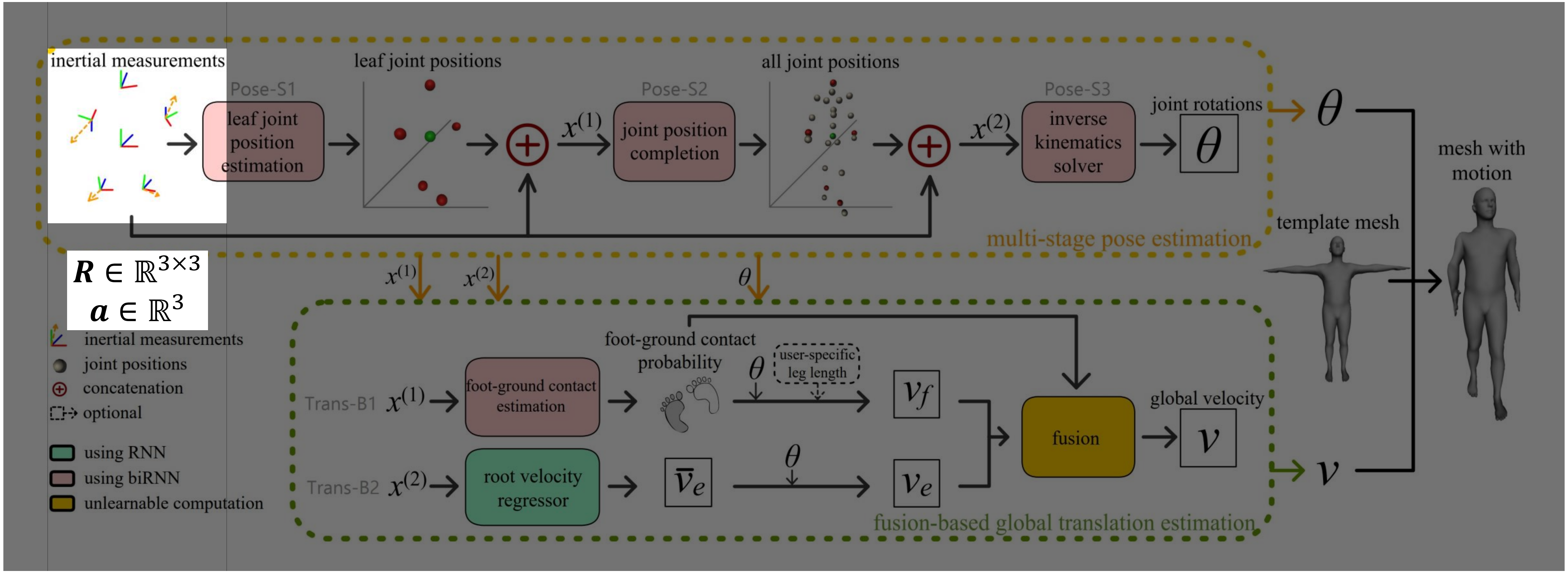
Method：TransPose

Overview of TransPose [2021]



Method： TransPose

Overview of TransPose [2021]



Input: orientations $\boldsymbol{R}$ and accelerations $\boldsymbol{a}$ of 6 IMUs

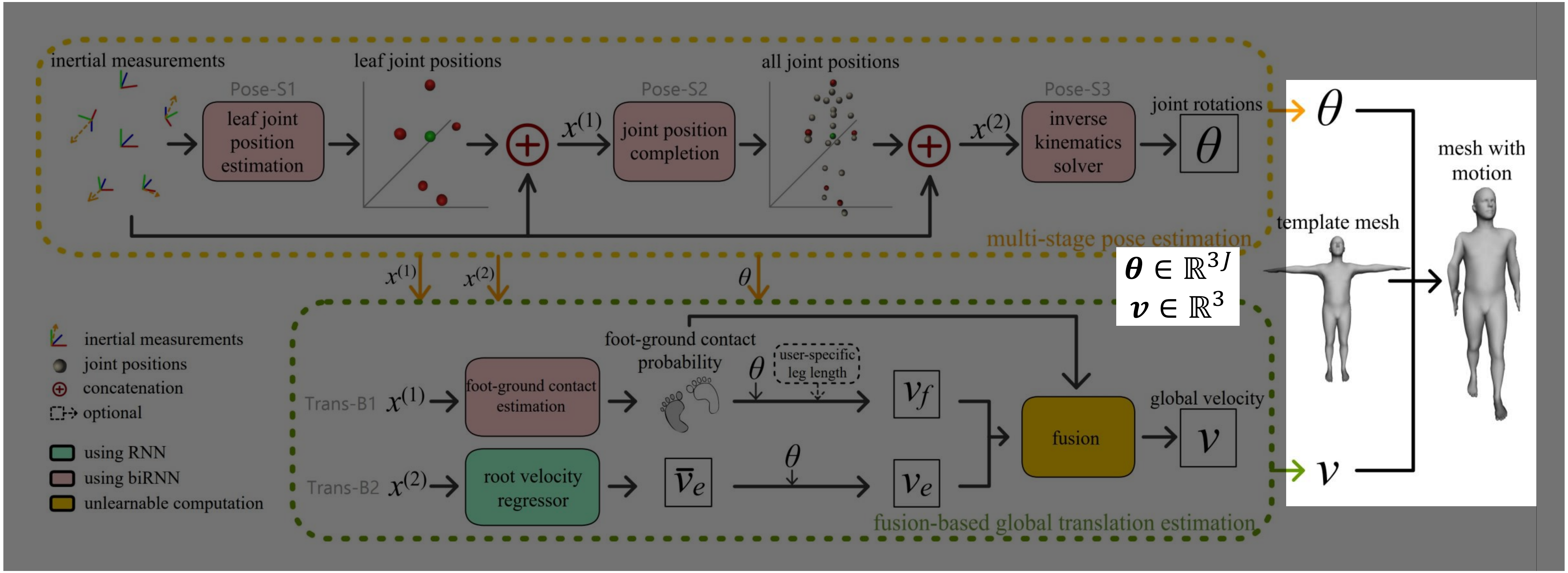
Method：TransPose

Overview of TransPose [2021]

Input: orientations $\boldsymbol{R}$ and accelerations $\boldsymbol{a}$ of 6 IMUs

Method: TransPose

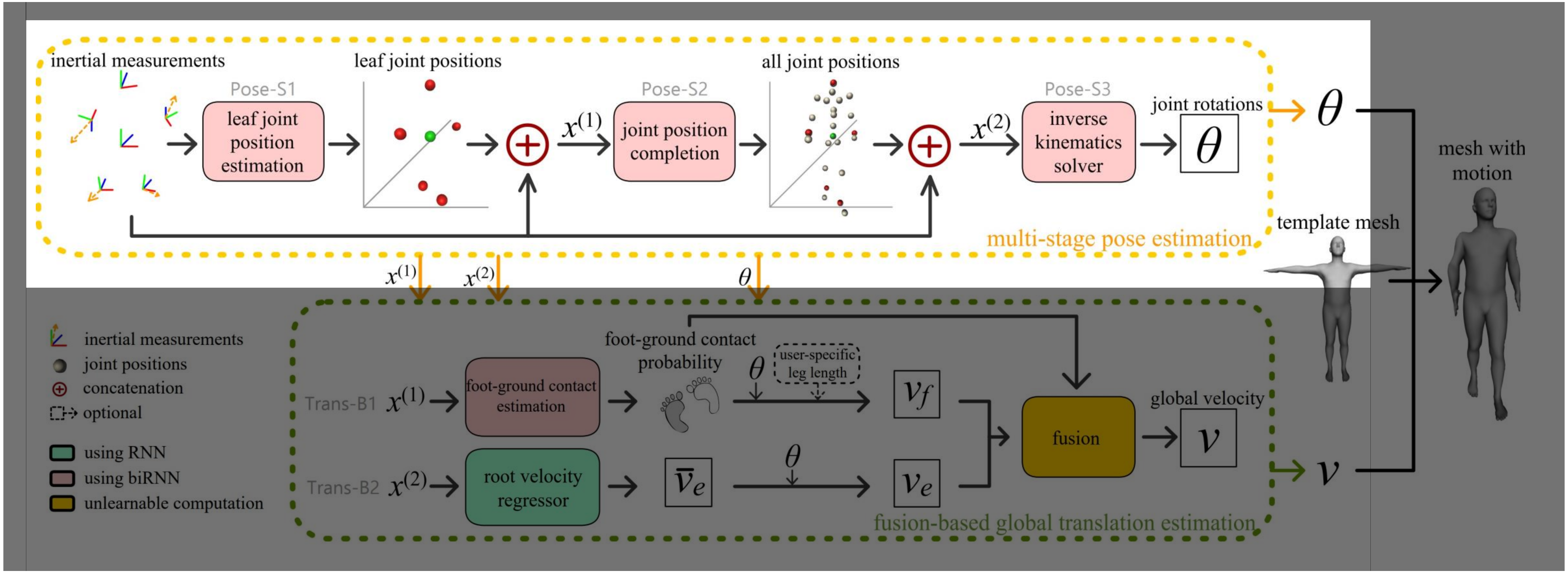
Overview of TransPose [2021]



Output: pose parameters $\boldsymbol{\theta}$ and translations $\boldsymbol{v}$ of the subject

Method：TransPose

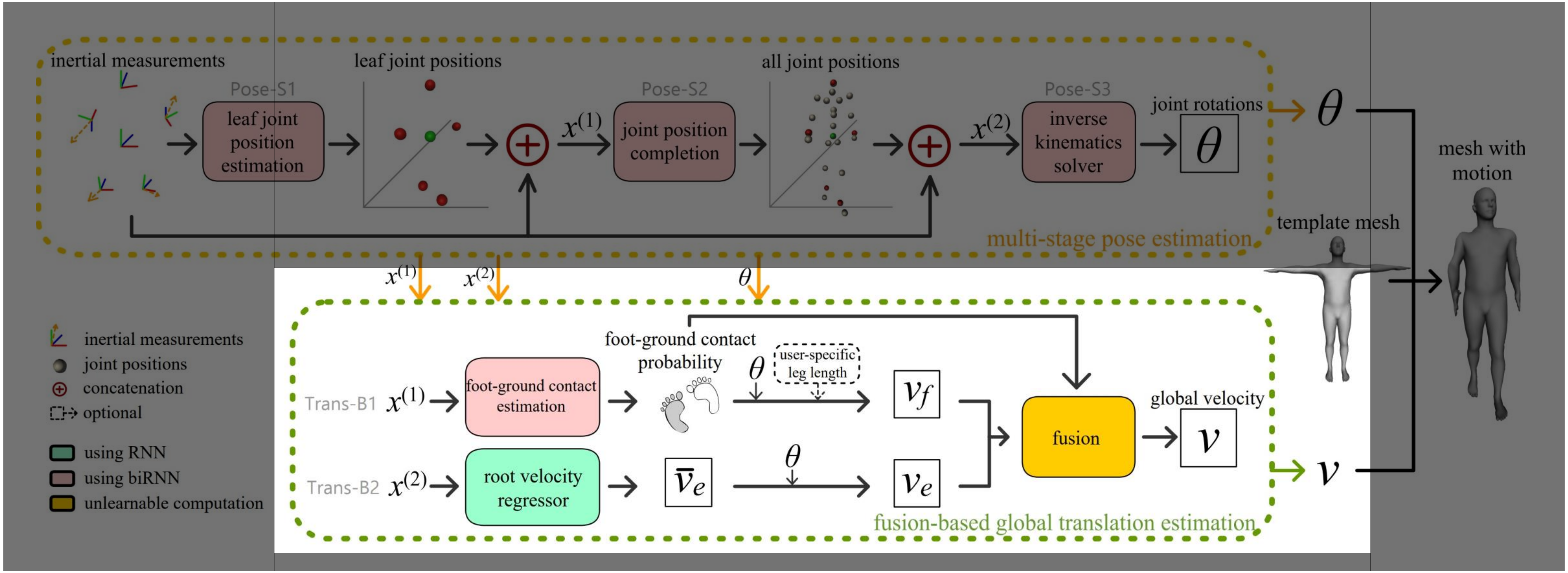
Overview of TransPose [2021]



Pose estimation subtask: pose parameters

Method：TransPose

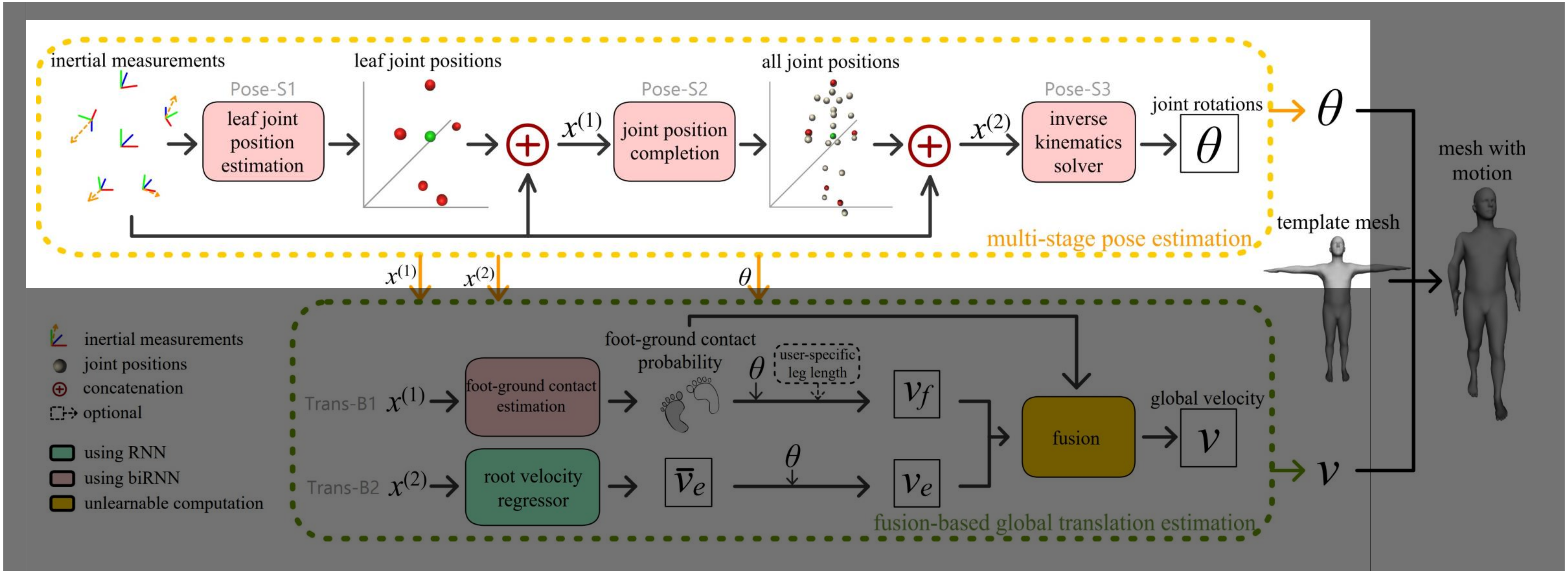
Overview of TransPose [2021]



Translation estimation subtask: global translations

Method：TransPose

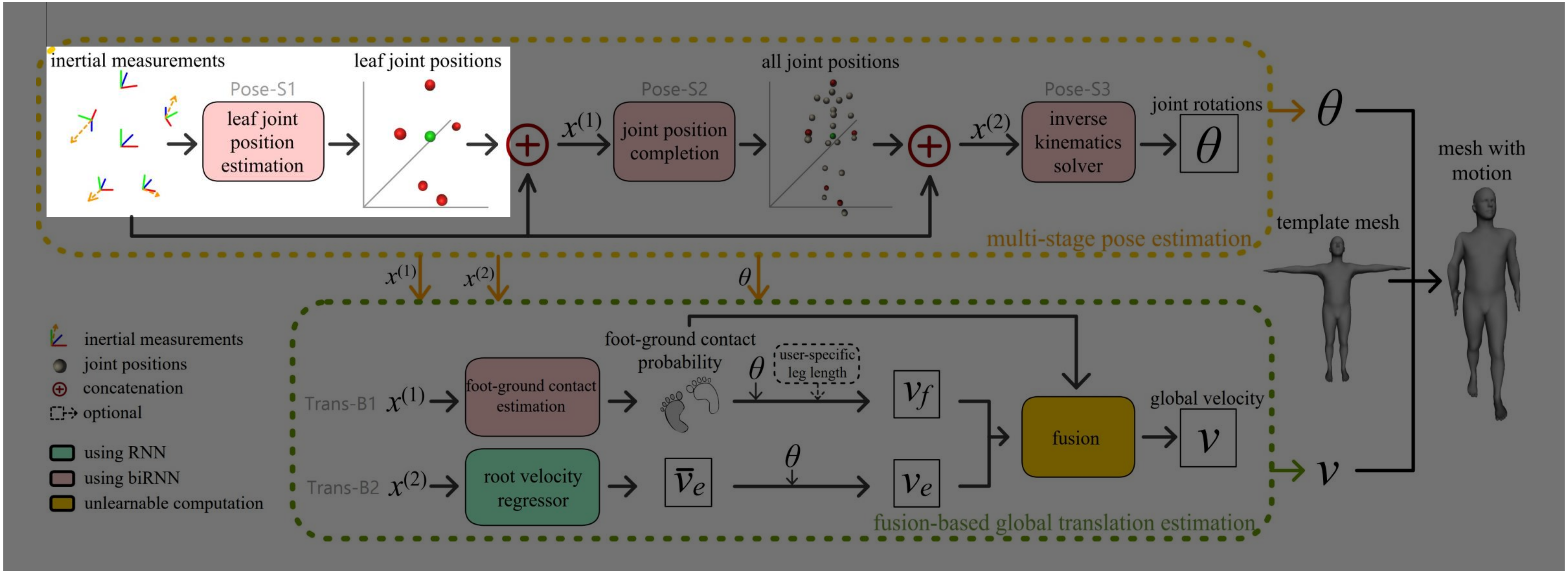
TransPose: multi-stage pose estimation



Introducing intermediate joint position estimation task to better model pose prior

Method：TransPose

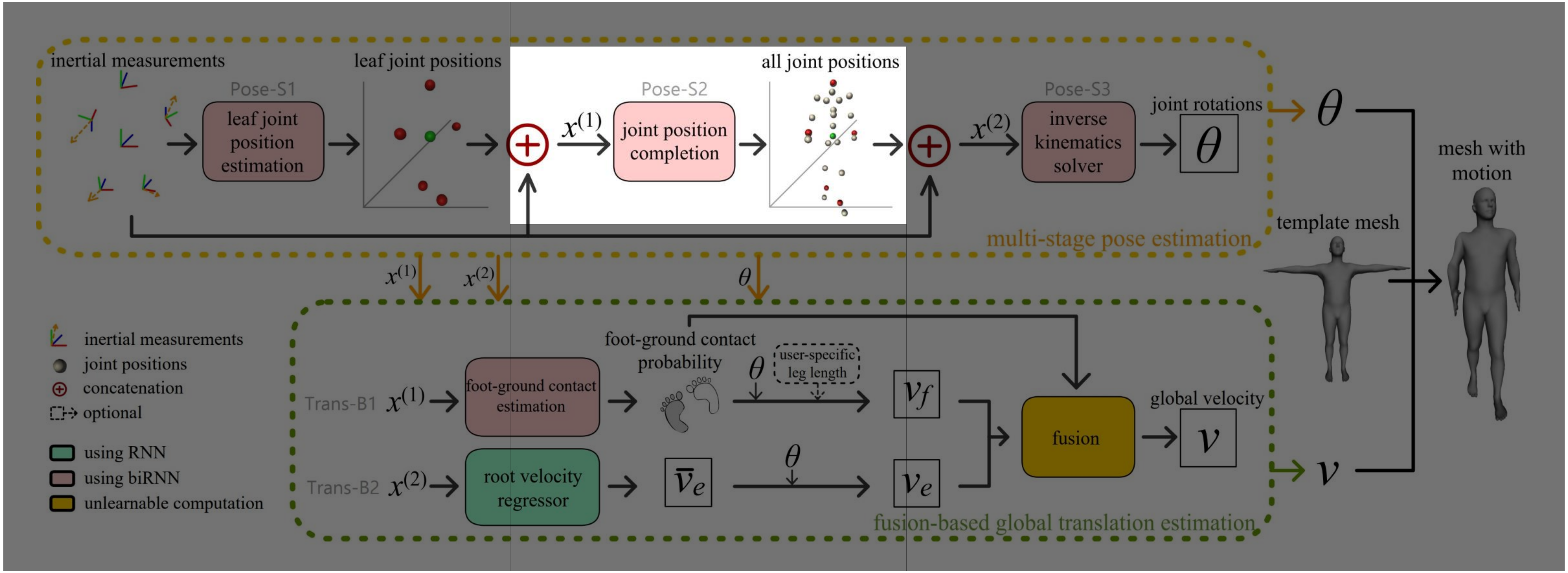
TransPose: multi-stage pose estimation



Pose Stage 1: IMUs → leaf joint positions

Method: TransPose

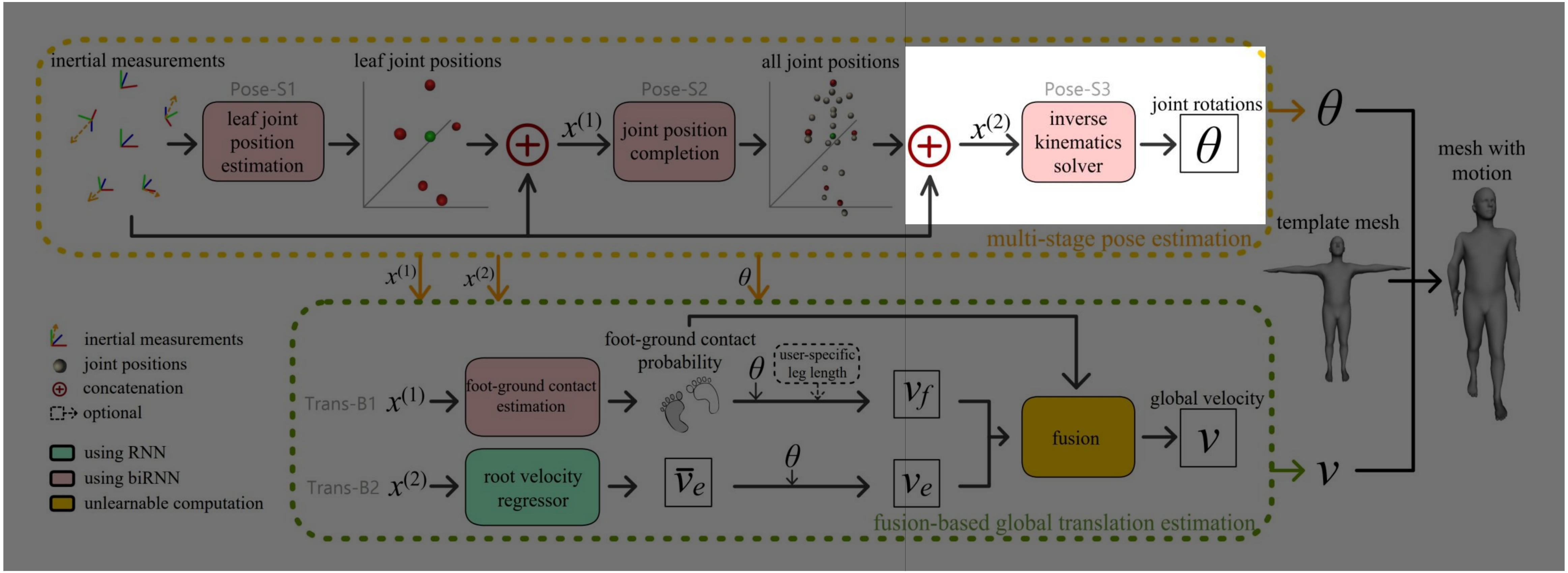
TransPose: multi-stage pose estimation



Pose Stage 2: IMUs + leaf joint positions → full joint positions

Method：TransPose

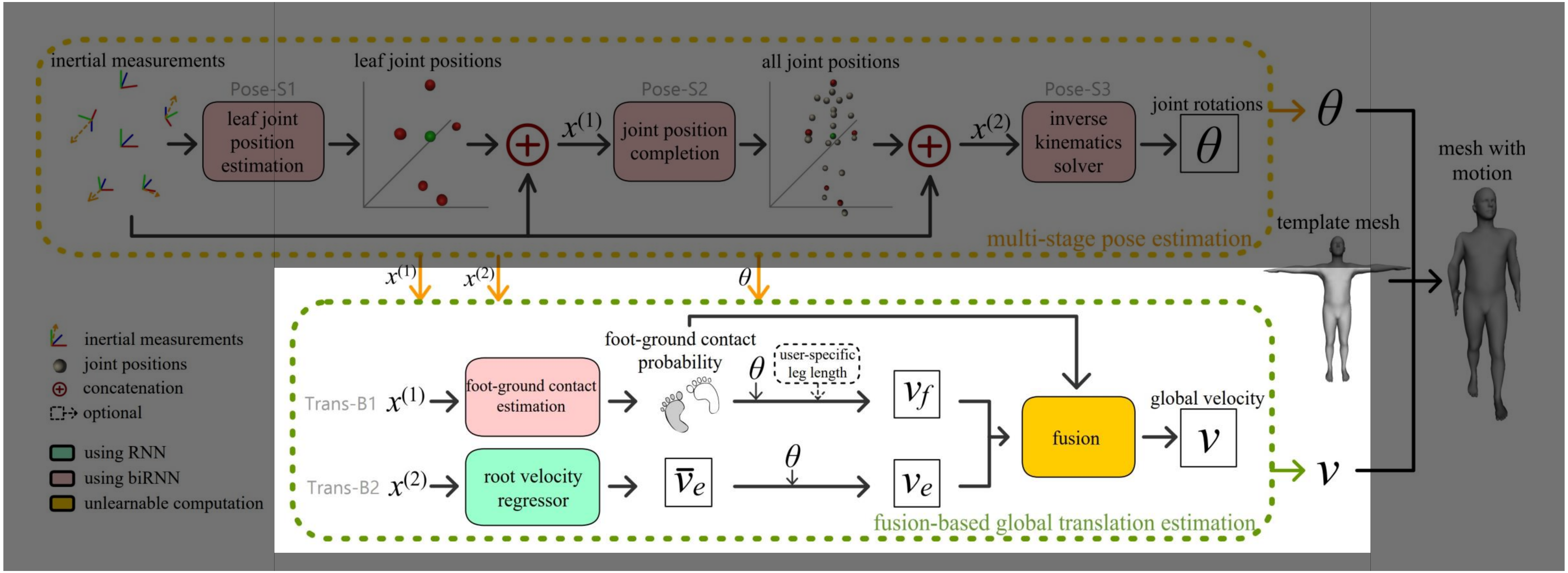
◻ TransPose: multi-stage pose estimation



Pose Stage 3: IMUs + full joint positions → joint rotations

Method：TransPose

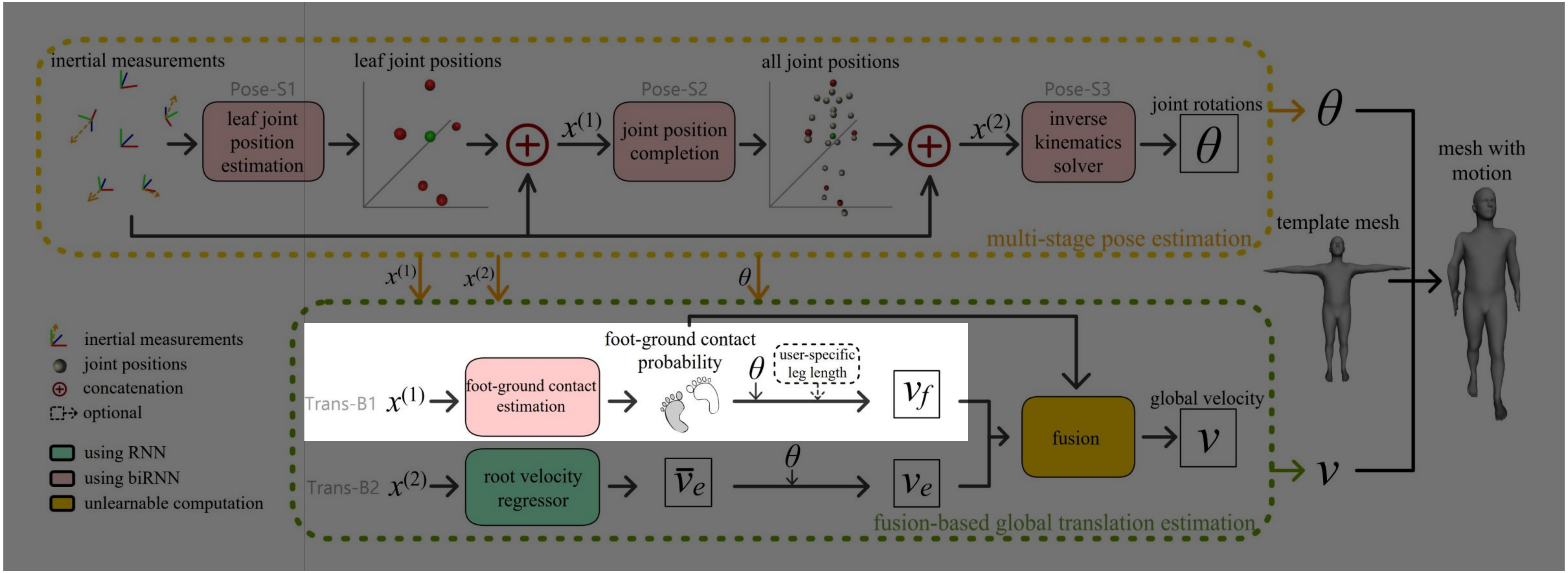
TransPose: fusion-based translation estimation



Leveraging physics rules and a complementary neural network to estimate translation

Method：TransPose

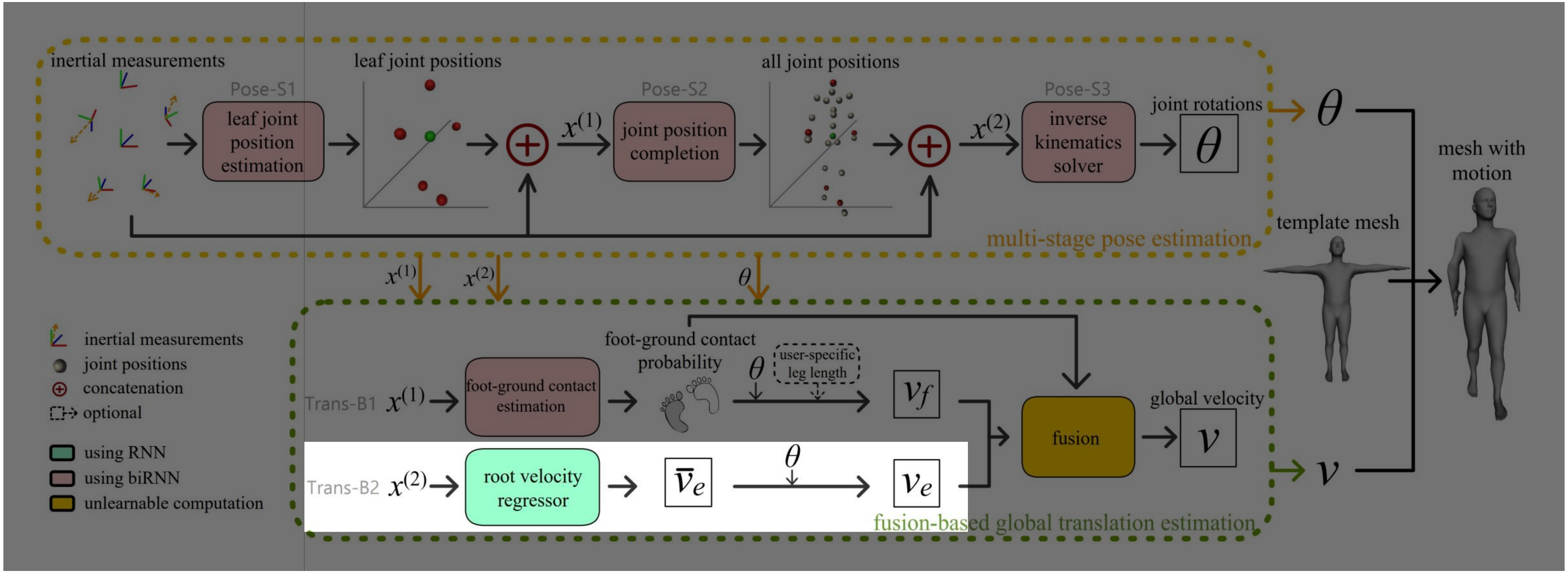
TransPose: fusion-based translation estimation



Translation Branch 1: IMUs + leaf joint positions → physics-rule-based translations

Method：TransPose

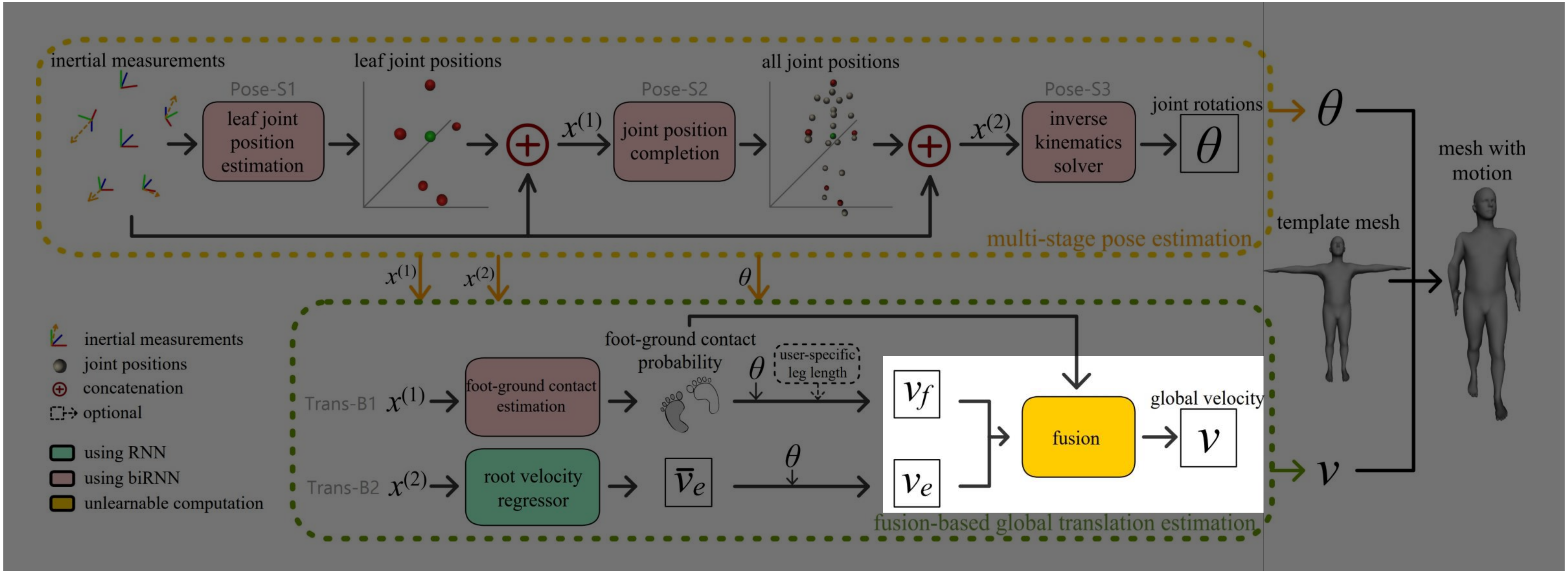
TransPose: fusion-based translation estimation



Translation Branch 2: IMUs + full joint positions → network-regressed translations

Method：TransPose

TransPose: fusion-based translation estimation



Translation Fusion: physics rule + network → final translation

Method：TransPose

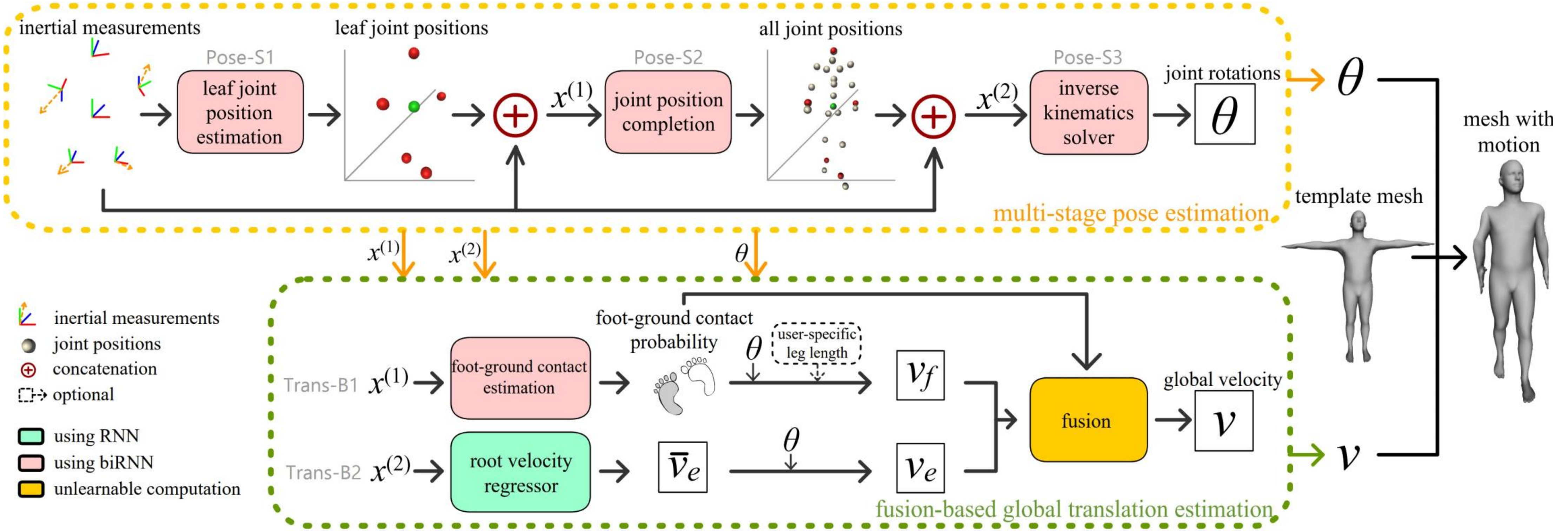
Supporting Foot Visualization I

supporting foot probability 0 1

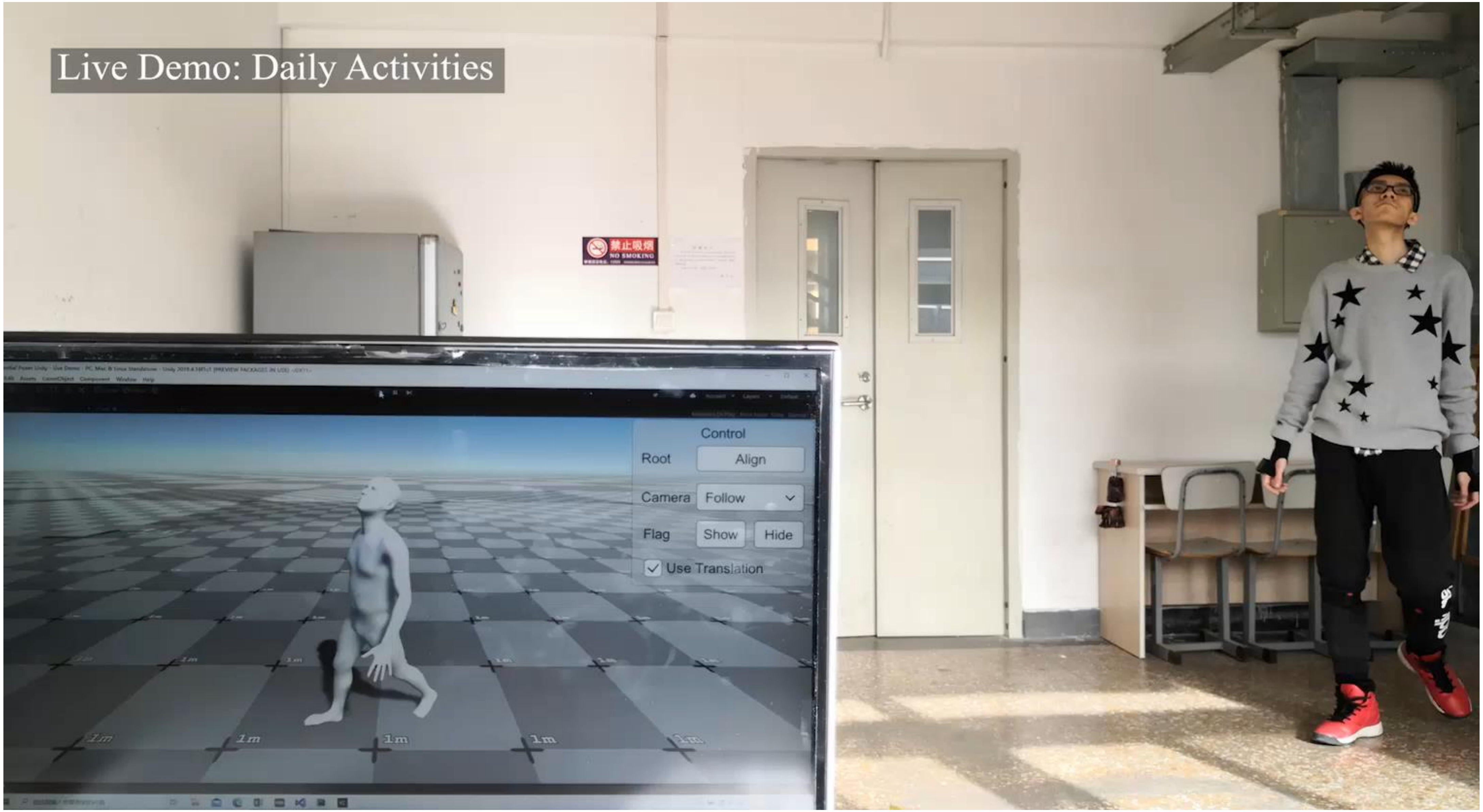
We record the sensor measurements and run our pipeline offline to render the supporting foot predictions.

Method：TransPose

Summary of TransPose [2021]



Method：TransPose



Method：PIP

- **Physical correctness in motion capture**

Kinematics

How does the subject move?

- Joint positions
- Joint rotations
- Joint velocities
- Joint accelerations
- …

vs

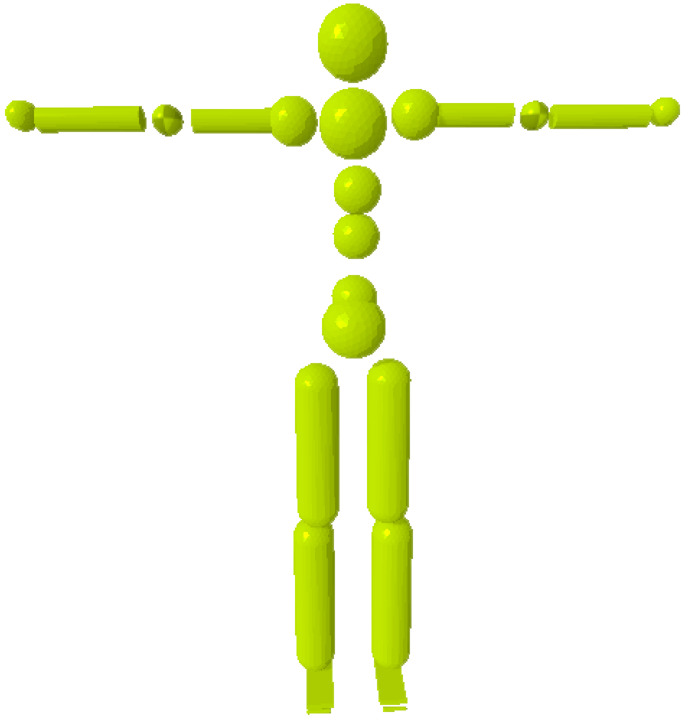
Dynamics

What causes the movement?

- Joint positions
- Joint rotations
- Joint velocities
- Joint accelerations
- **Joint torques**
- **Body mass distribution**
- **Body shapes**
- **Contact forces**
- …

Method：PIP

Physics model for human body



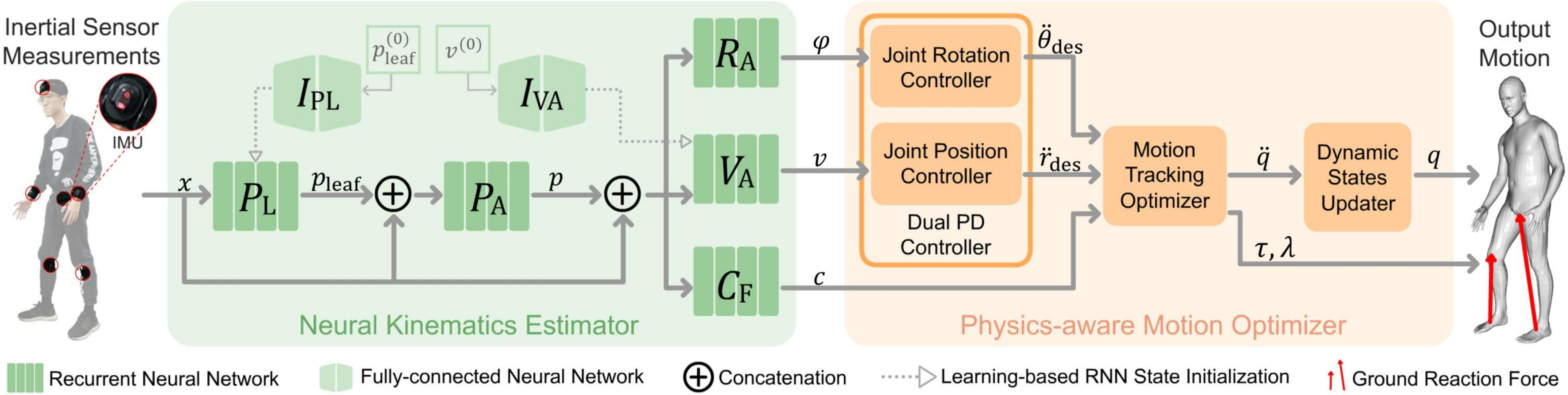
Just like a …

We use a torque-controlled floating-base character model

based on PhysCap [Shimada et al. 2020]

Method： PIP

Our system



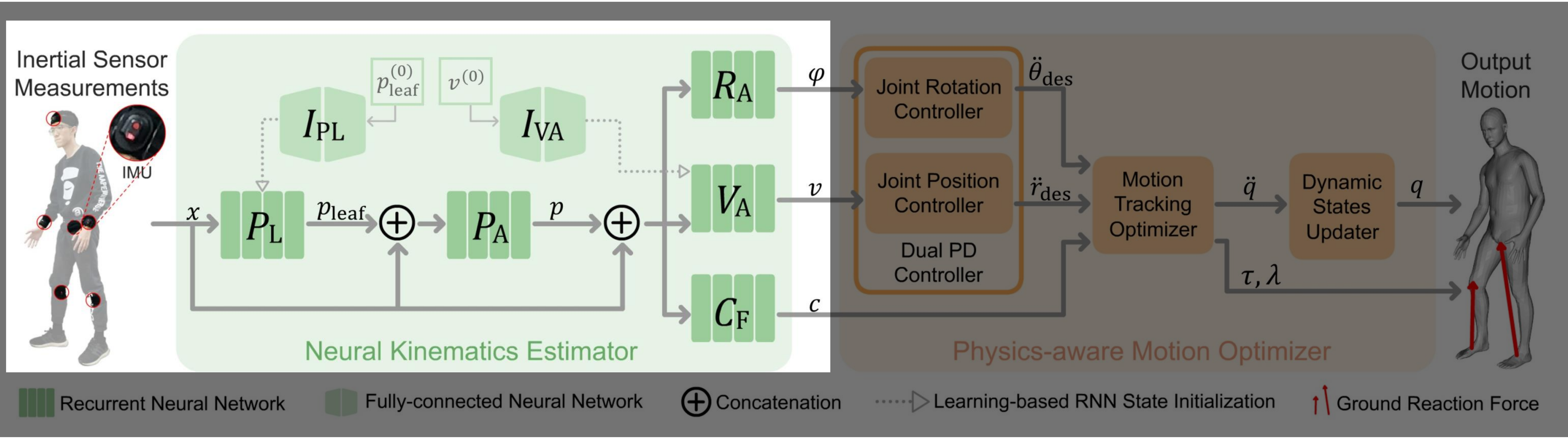
Our system consists of a neural kinematics estimator and a physics-based motion optimizer

Model motion prior

Model physics rules

Method：PIP

Kinematics Estimator: inertia measurements → motion status

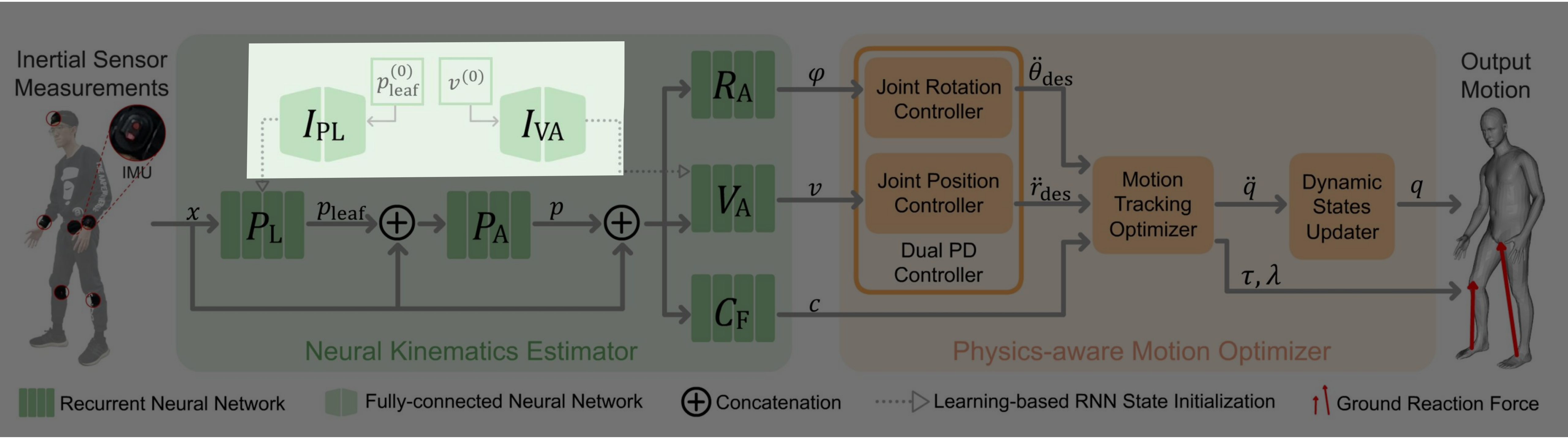


Different from TransPose [Yi et al. 2021], we use unidirectional LSTM

- to retain full historical information during online prediction
- for better runtime performance and lower latency

Method: PIP

- **Kinematics Estimator: inertia measurements → motion status**

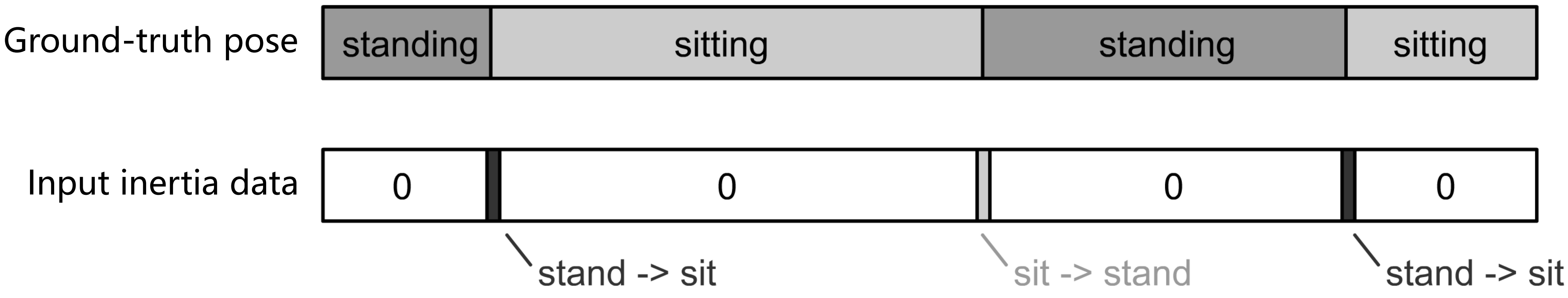


To disambiguate motions with similar sensor measurements, only using RNN is not enough …

We need a new RNN hidden state initialization scheme

Method：PIP

Typical RNN training

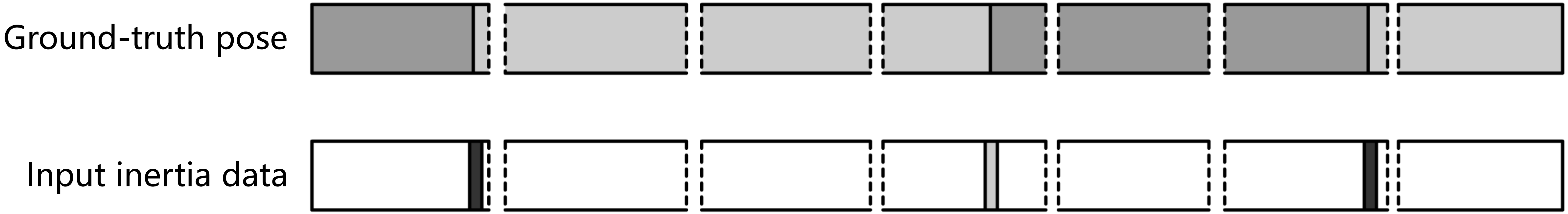


We use stand/sit as an example

(ambiguity comes from similar IMU measurements)

Method： PIP

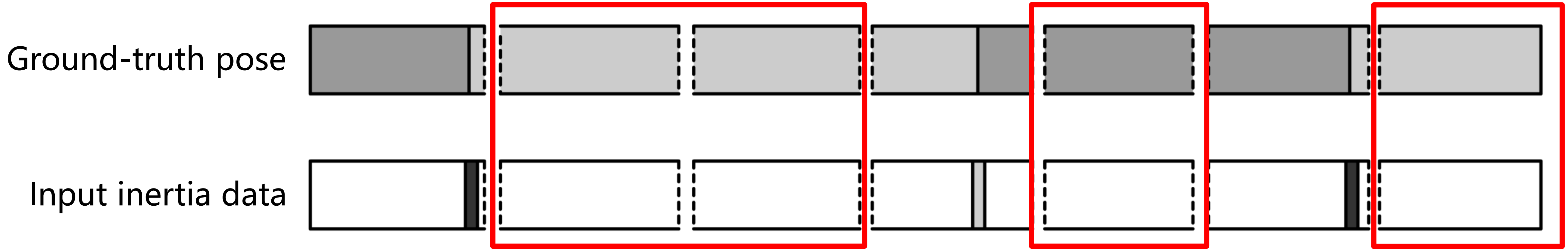
Typical RNN training



Typically, we cut the input sequences into small pieces
and train RNNs in a mini-batch manner

Method： PIP

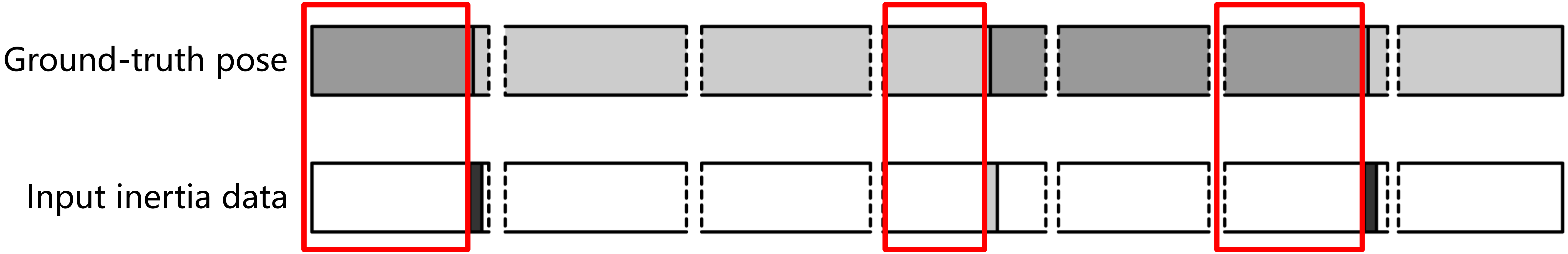
Typical RNN training



Oops! The network is trained with the same inputs but different outputs

Method： PIP

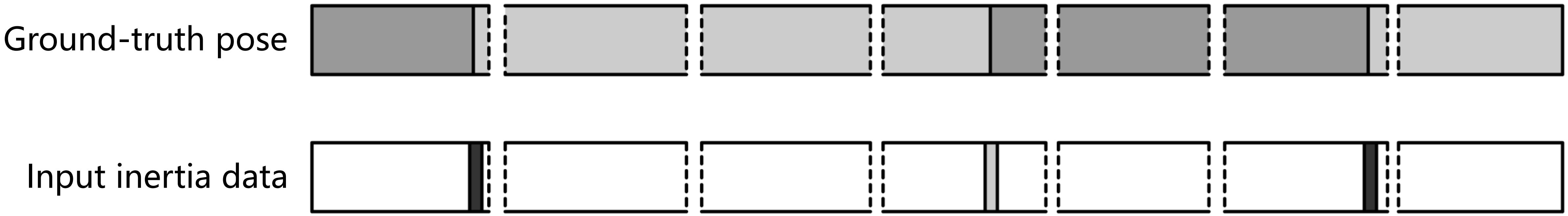
Typical RNN training



On the other pieces, the first few frames are also inconsistently trained!

Method：PIP

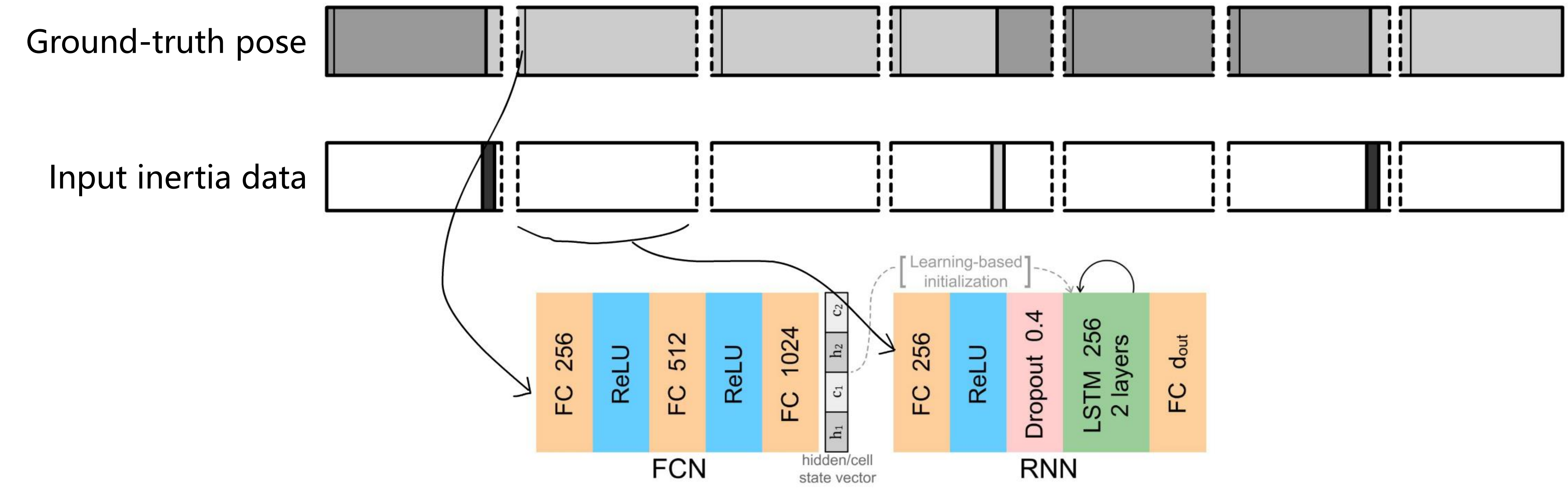
Learning-based RNN initialization



The problem is, the RNN hidden states are always constantly initialized (e.g., zero), while the beginning pose of each sequence can be different (e.g., standing/sitting/lying)

Method： PIP

Learning-based RNN initialization

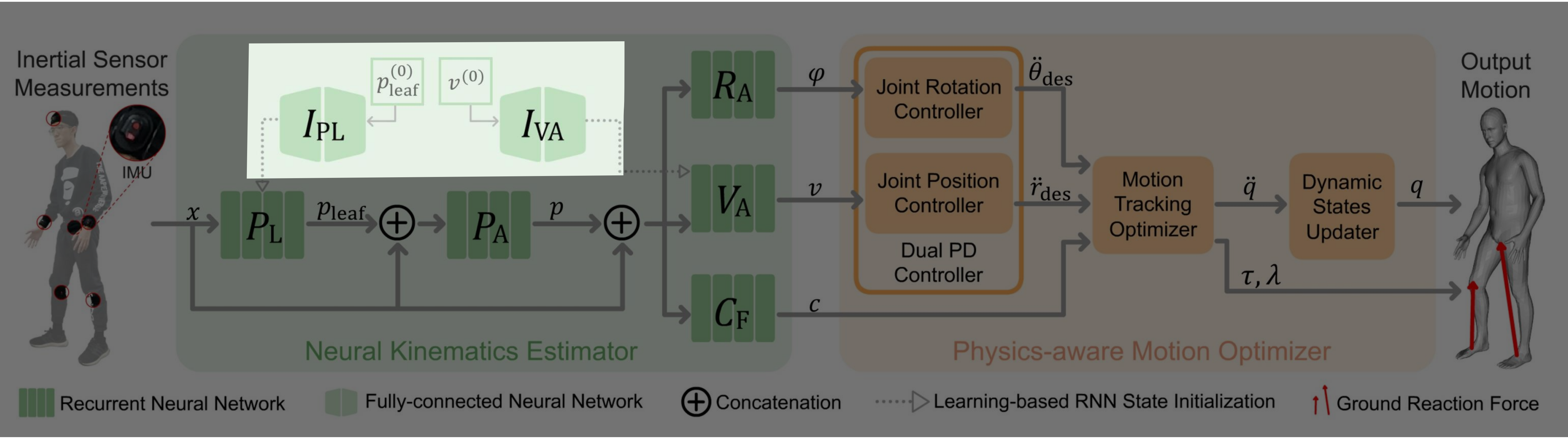


During training, we regress the RNN hidden state from the beginning pose

The RNN is trained as usual (compatible with black-box RNN implementation)

Method: PIP

Kinematics Estimator: inertia measurements → motion status



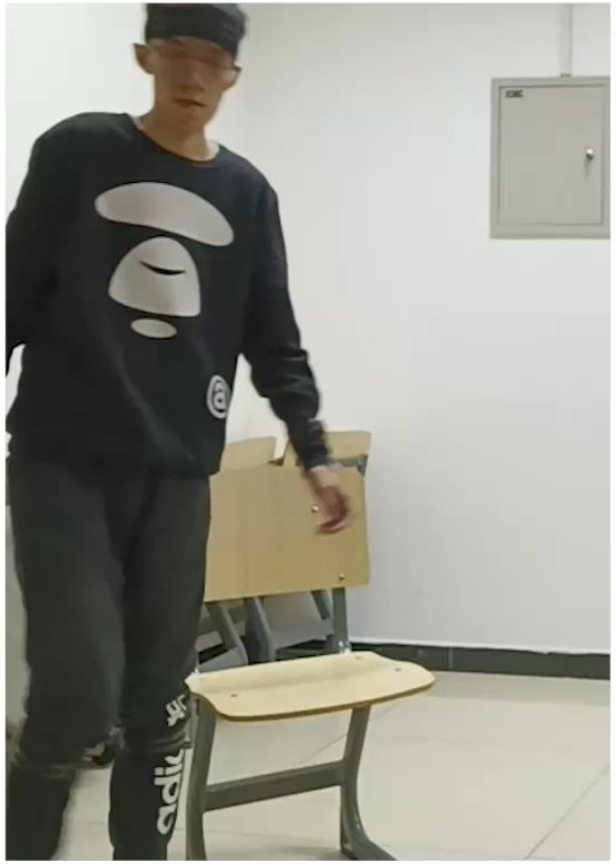
During prediction, the subject will always begin with T-pose as we need a T-pose calibration

These two networks are only used at the beginning of the capture

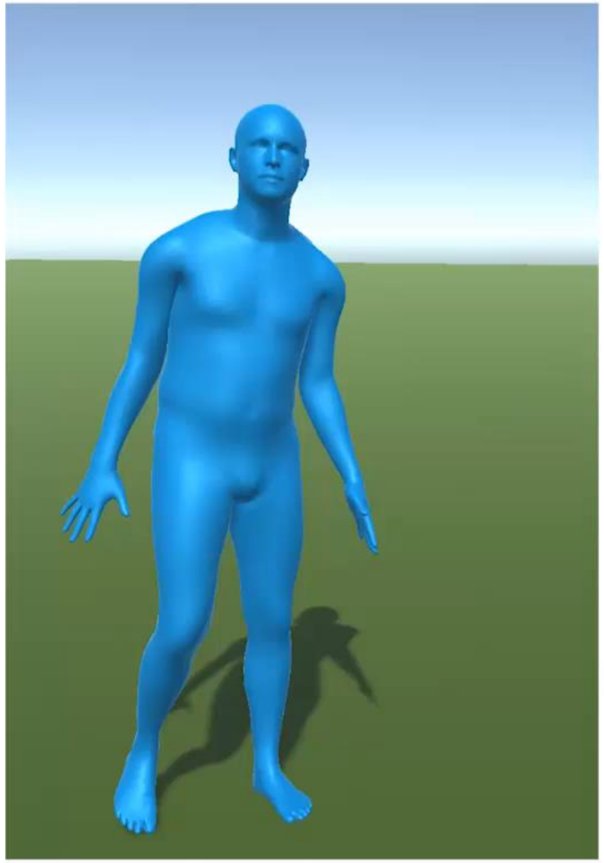
Method： PIP

Evaluation: Learning-based Initialization

Our learning-based RNN initialization technique helps to resolve the pose ambiguity.



Video Reference (not input)



w/o Learning-based Initialization

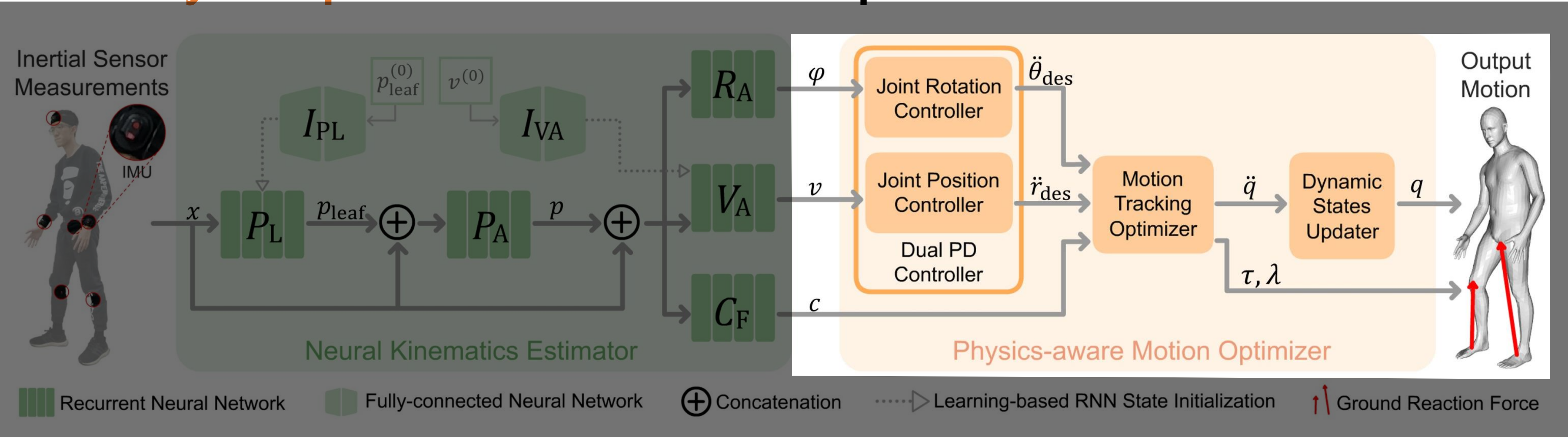
Ours

Method: PIP

We propose a novel RNN initialization scheme which helps with pose disambiguation

Method： PIP

- **Physics Optimizer : motion status → pose & translation & forces**

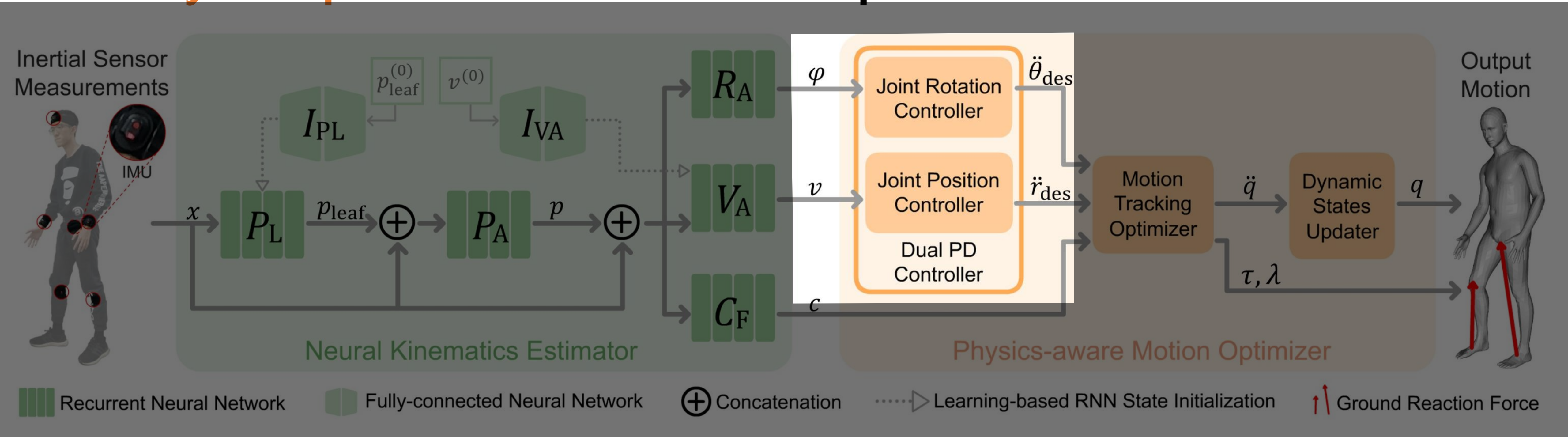


Target: estimate a set of forces & torques to control the physics model

to imitate the kinematically predicted motion

Method: PIP

- **Physics Optimizer** : motion status → pose & translation & forces

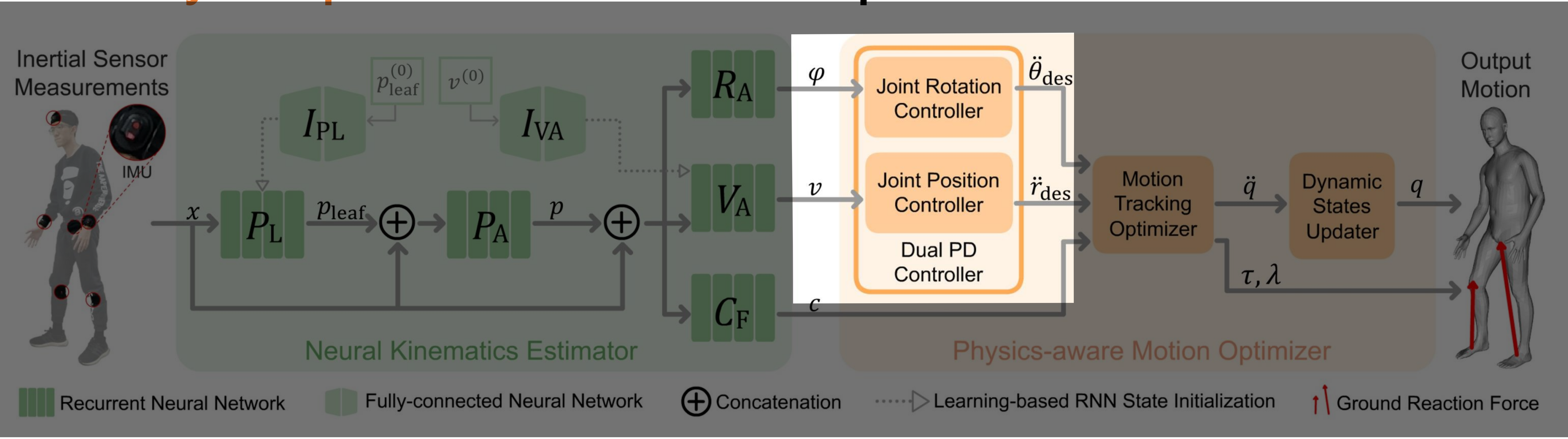


To imitate the reference motion, what accelerations does the character need?

$\ddot{\theta}_{des} = k_{p_\theta}(\theta_{ref} - \theta) - k_{d_\theta}\dot{\theta}$ ← Control the local rotation of each joint

Method: PIP

- **Physics Optimizer : motion status → pose & translation & forces**



To imitate the reference motion, what accelerations does the character need?

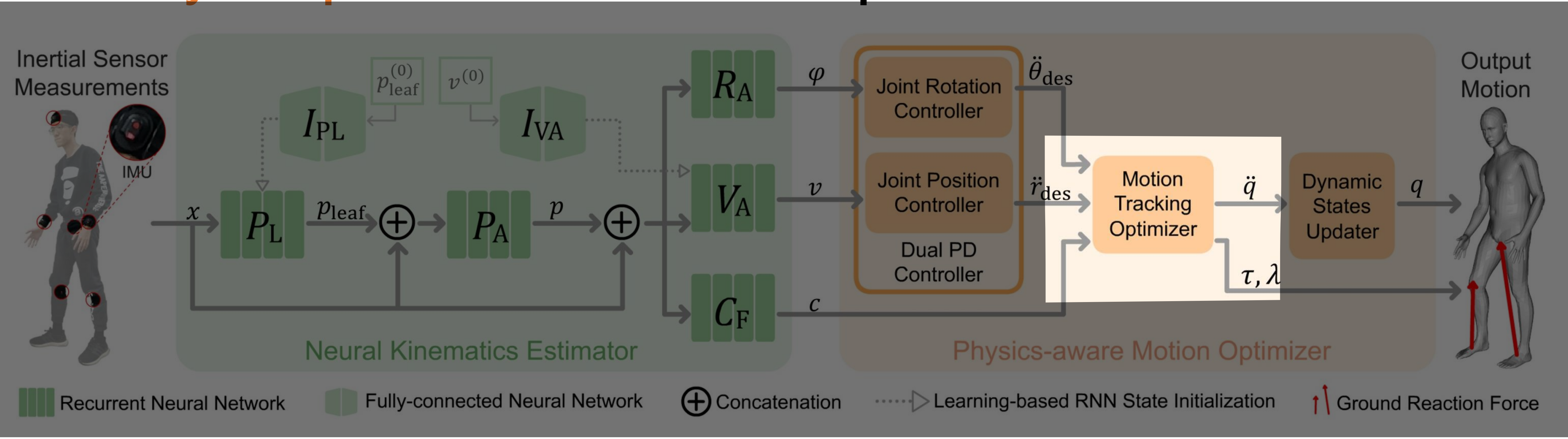
$\ddot{\theta}_{des} = k_{p_\theta}(\theta_{ref} - \theta) - k_{d_\theta}\dot{\theta}$ ← Control the local rotation of each joint

$\ddot{r}_{des} = k_{p_r}(r_{ref} - r) - k_{d_r}\dot{r}$ ← Control the global rotation of each joint

$r_{ref} = r + T(v)\Delta t, \quad \dot{r} = J\dot{q}$

Method： PIP

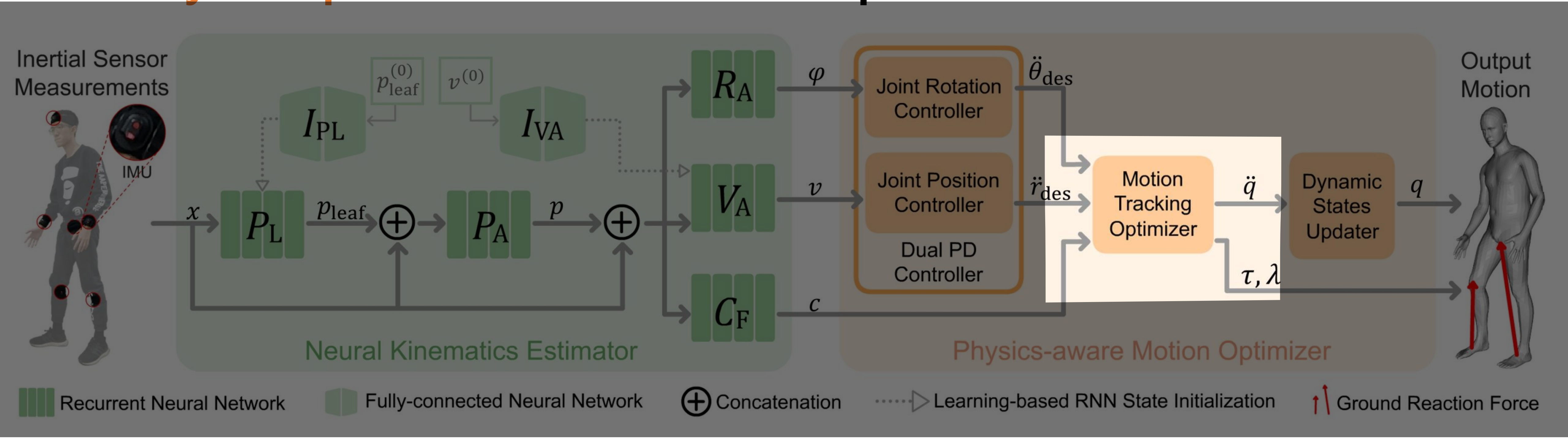
◻ Physics Optimizer : motion status → pose & translation & forces



$$
\begin{aligned}
\underset{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{\tau}}{\arg\min} \quad & \mathcal{E}_{\mathrm{PD}} + \mathcal{E}_{\mathrm{reg}} \\
\text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{c}}^T \boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} \quad && \text{(equation of motion)} \\
& \boldsymbol{\lambda} \in \mathcal{F} && \text{(friction cone)} \\
& \dot{\boldsymbol{r}}_{\mathrm{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} && \text{(no sliding).}
\end{aligned}
$$

Method： PIP

- **Physics Optimizer : motion status → pose & translation & forces**



$$
\begin{aligned}
\boxed{\underset{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{\tau}}{\arg\min}} \quad & \mathcal{E}_{\mathrm{PD}} + \mathcal{E}_{\mathrm{reg}} \\
\text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{c}}^{T}\boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} && \text{(equation of motion)} \\
& \boldsymbol{\lambda} \in \mathcal{F} && \text{(friction cone)} \\
& \dot{\boldsymbol{r}}_{\mathrm{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} && \text{(no sliding).}
\end{aligned}
$$

Quadratic Programming

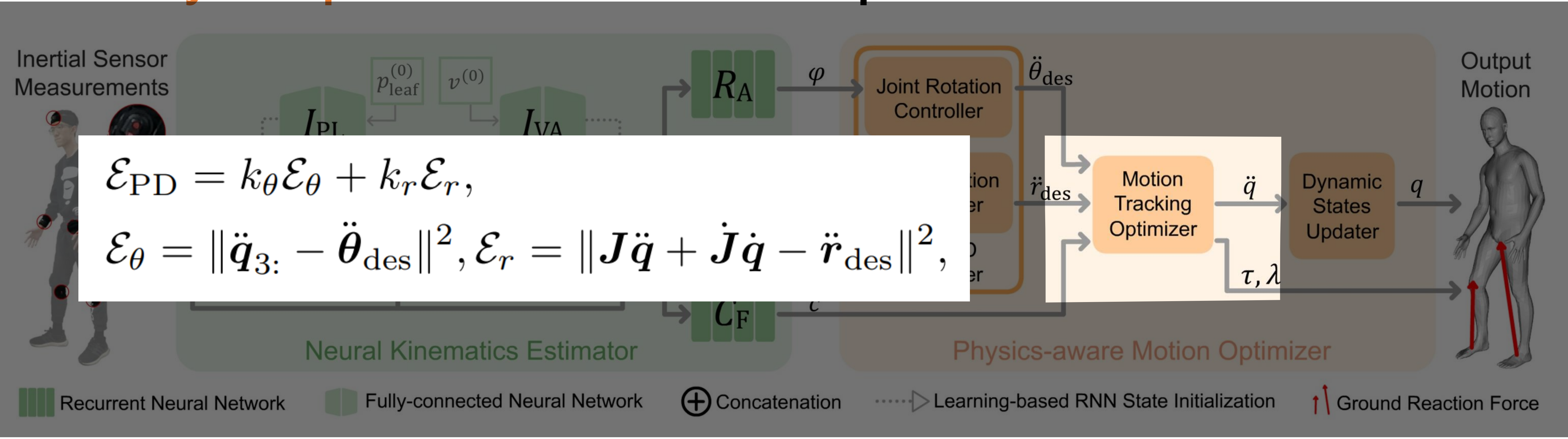
$\ddot{q}$: acceleration

λ: ground reaction forces

τ: joint torques

Method：PIP

- **Physics Optimizer : motion status → pose & translation & forces**



$$\mathcal{E}_{\mathrm{PD}} = k_\theta \mathcal{E}_\theta + k_r \mathcal{E}_r,$$

$$\mathcal{E}_\theta = \|\ddot{\boldsymbol{q}}_{3:} - \ddot{\boldsymbol{\theta}}_{\mathrm{des}}\|^2, \mathcal{E}_r = \|\boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \ddot{\boldsymbol{r}}_{\mathrm{des}}\|^2,$$

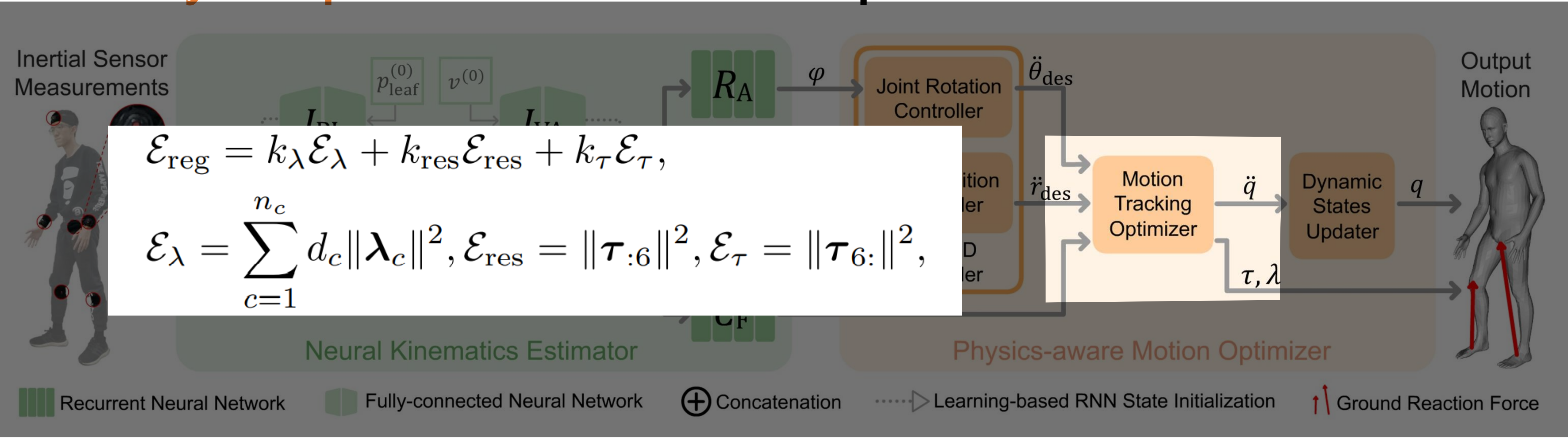
$$\begin{aligned} \underset{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{\tau}}{\arg\min} \quad & \boxed{\mathcal{E}_{\mathrm{PD}}} + \mathcal{E}_{\mathrm{reg}} \\ \text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{c}}^T \boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} && \text{(equation of motion)} \\ & \boldsymbol{\lambda} \in \mathcal{F} && \text{(friction cone)} \\ & \dot{\boldsymbol{r}}_{\mathrm{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} && \text{(no sliding).} \end{aligned}$$

PD term $\mathcal{E}_{\mathrm{PD}}(\ddot{q})$:

The linear & angular accelerations should be similar to the ones given by the dual PD controller

Method: PIP

Physics Optimizer : motion status → pose & translation & forces



$$\mathcal{E}_{\mathrm{reg}} = k_\lambda \mathcal{E}_\lambda + k_{\mathrm{res}} \mathcal{E}_{\mathrm{res}} + k_\tau \mathcal{E}_\tau,$$

$$\mathcal{E}_\lambda = \sum_{c=1}^{n_c} d_c \|\boldsymbol{\lambda}_c\|^2, \mathcal{E}_{\mathrm{res}} = \|\boldsymbol{\tau}_{:6}\|^2, \mathcal{E}_\tau = \|\boldsymbol{\tau}_{6:}\|^2,$$

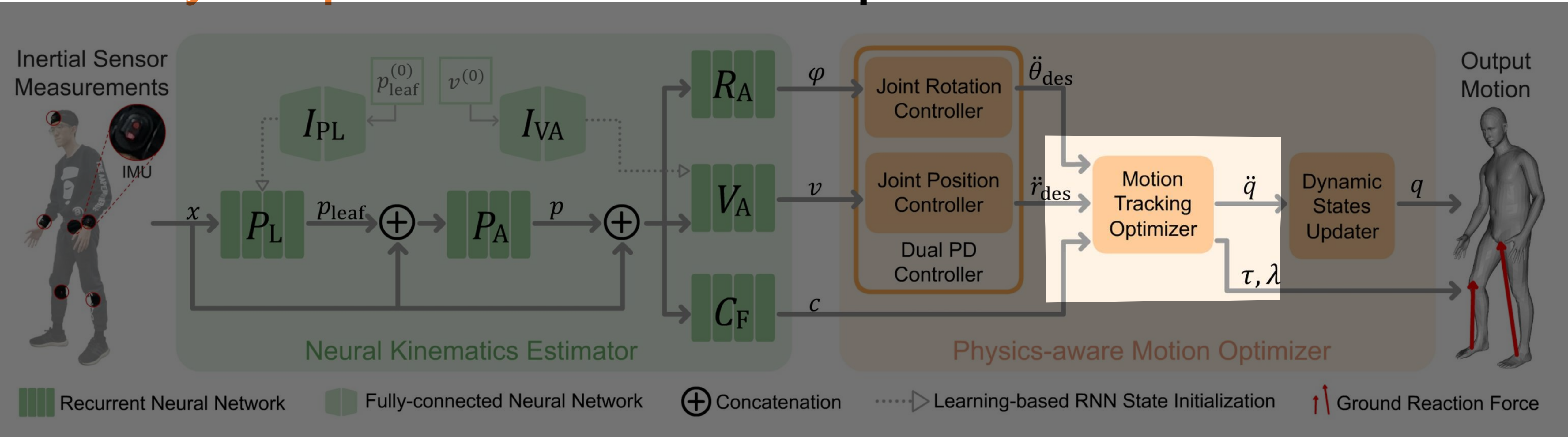
$$\begin{aligned} \underset{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{\tau}}{\arg\min} \quad & \mathcal{E}_{\mathrm{PD}} + \boxed{\mathcal{E}_{\mathrm{reg}}} \\ \text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{c}}^T \boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} && \text{(equation of motion)} \\ & \boldsymbol{\lambda} \in \mathcal{F} && \text{(friction cone)} \\ & \dot{\boldsymbol{r}}_{\mathrm{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} && \text{(no sliding).} \end{aligned}$$

Regularization term $\mathcal{E}_{\mathrm{reg}}(\tau, \lambda)$:

- External forces must act on contacts
- Joint torques should be small
- No actuation at the root joint

Method： PIP

Physics Optimizer : motion status → pose & translation & forces



$$
\begin{aligned}
\underset{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{\tau}}{\arg\min} \quad & \mathcal{E}_{\mathrm{PD}} + \mathcal{E}_{\mathrm{reg}} \\
\text{s.t.} \quad & \boxed{\boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{c}}^{T}\boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} \qquad \text{(equation of motion)}} \\
& \boldsymbol{\lambda} \in \mathcal{F} \qquad \text{(friction cone)} \\
& \dot{\boldsymbol{r}}_{\mathrm{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} \qquad \text{(no sliding).}
\end{aligned}
$$

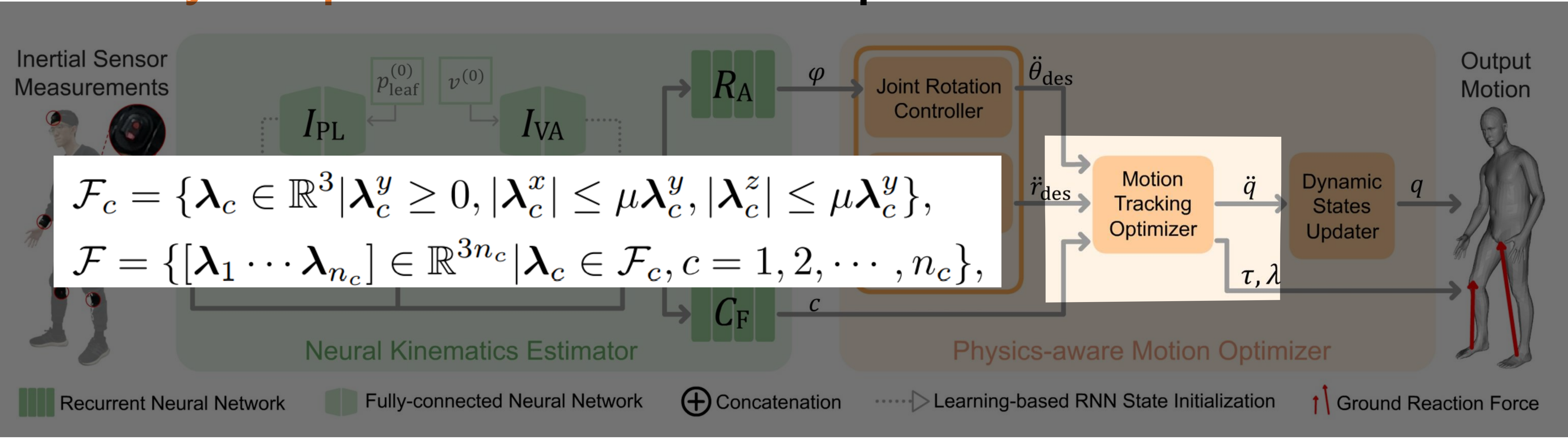
Equation of motion:

The relationship between forces and accelerations —— in other words,

$F = ma$

Method： PIP

□ Physics Optimizer : motion status → pose & translation & forces



$$\mathcal{F}_c = \{\boldsymbol{\lambda}_c \in \mathbb{R}^3 | \boldsymbol{\lambda}_c^y \geq 0, |\boldsymbol{\lambda}_c^x| \leq \mu \boldsymbol{\lambda}_c^y, |\boldsymbol{\lambda}_c^z| \leq \mu \boldsymbol{\lambda}_c^y\},$$
$$\mathcal{F} = \{[\boldsymbol{\lambda}_1 \cdots \boldsymbol{\lambda}_{n_c}] \in \mathbb{R}^{3n_c} | \boldsymbol{\lambda}_c \in \mathcal{F}_c, c = 1, 2, \cdots, n_c\},$$

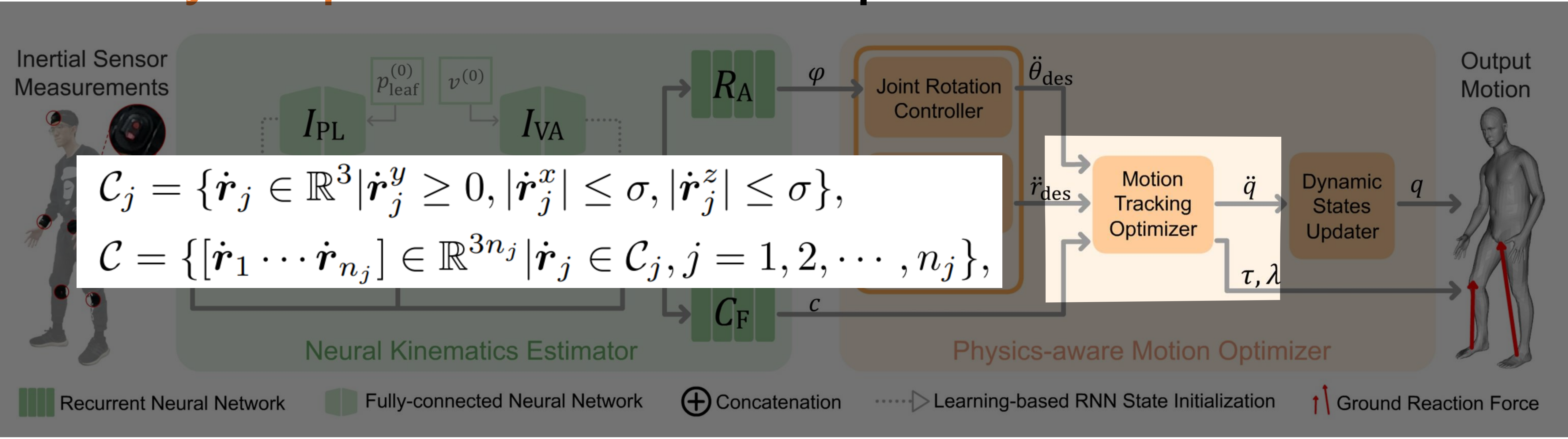
$$\begin{aligned}
\arg\min_{\ddot{\boldsymbol{q}}, \boldsymbol{\lambda}, \boldsymbol{\tau}} \quad & \mathcal{E}_{\text{PD}} + \mathcal{E}_{\text{reg}} & \\
\text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_c^T \boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} & \text{(equation of motion)} \\
& \boxed{\boldsymbol{\lambda} \in \mathcal{F} \qquad \text{(friction cone)}} & \\
& \dot{\boldsymbol{r}}_{\text{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} & \text{(no sliding).}
\end{aligned}$$

Friction cone:

Linearization of the Coulomb friction law —— in other words, $f \leq \mu N$ and $N \geq 0$

Method: PIP

Physics Optimizer : motion status → pose & translation & forces



$$\mathcal{C}_j = \{\dot{\boldsymbol{r}}_j \in \mathbb{R}^3 | \dot{\boldsymbol{r}}_j^y \geq 0, |\dot{\boldsymbol{r}}_j^x| \leq \sigma, |\dot{\boldsymbol{r}}_j^z| \leq \sigma\},$$

$$\mathcal{C} = \{[\dot{\boldsymbol{r}}_1 \cdots \dot{\boldsymbol{r}}_{n_j}] \in \mathbb{R}^{3n_j} | \dot{\boldsymbol{r}}_j \in \mathcal{C}_j, j = 1, 2, \cdots, n_j\},$$

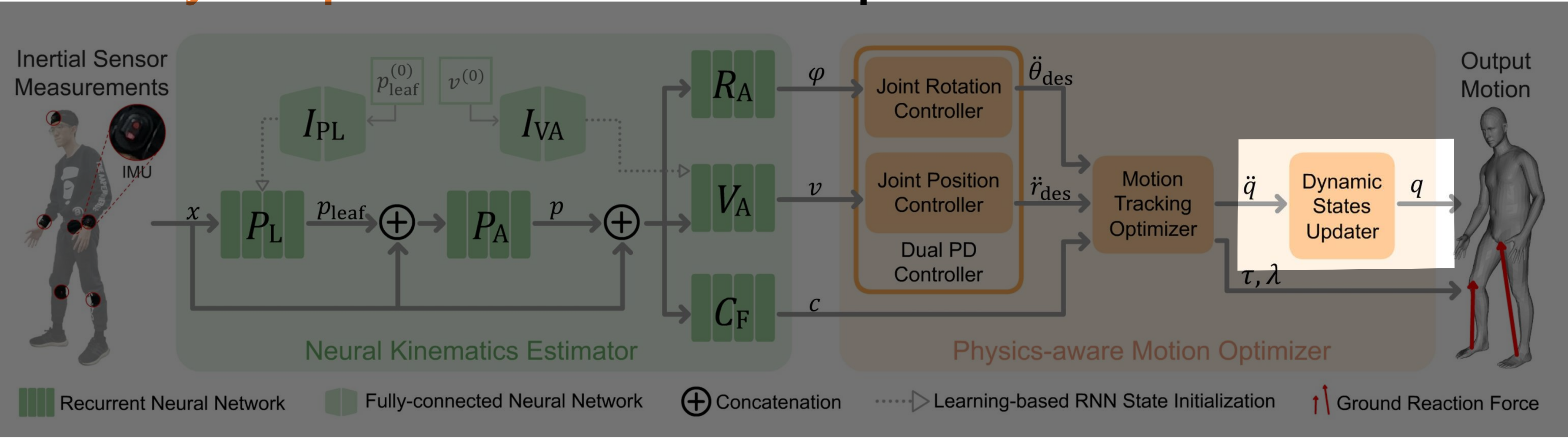
$$\begin{aligned} \underset{\ddot{\boldsymbol{q}}, \boldsymbol{\lambda}, \boldsymbol{\tau}}{\arg\min} \quad & \mathcal{E}_{\text{PD}} + \mathcal{E}_{\text{reg}} \\ \text{s.t.} \quad & \boldsymbol{\tau} + \boldsymbol{J}_{\text{c}}^T \boldsymbol{\lambda} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} \quad \text{(equation of motion)} \\ & \boldsymbol{\lambda} \in \mathcal{F} \quad \text{(friction cone)} \\ & \boxed{\dot{\boldsymbol{r}}_{\text{j}}(\ddot{\boldsymbol{q}}) \in \mathcal{C} \quad \text{(no sliding).}} \end{aligned}$$

No sliding:

The body should not slide or penetrate the ground at the contact points

Method: PIP

- **Physics Optimizer** : motion status → pose & translation & forces



Update the pose & translation from the estimated acceleration

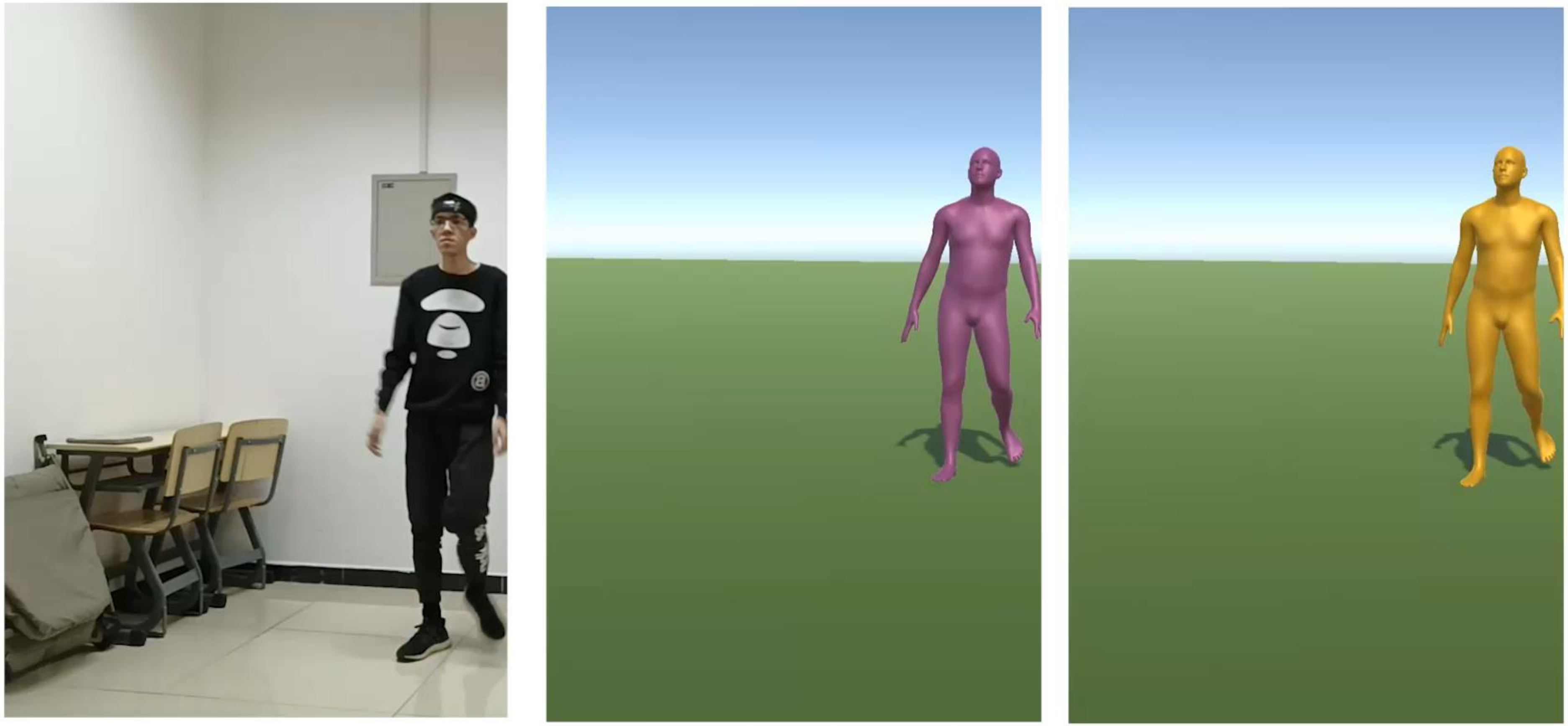
$$q^{(t+1)} = q^{(t)} + \dot{q}^{(t)}\Delta t$$

$$\dot{q}^{(t+1)} = \dot{q}^{(t)} + \ddot{q}^{(t)}\Delta t$$

Method： PIP

Evaluation: Dual PD Controller

Our dual proportional-derivative controller helps to improve physical plausibility.



Video Reference
(not input)

w/o Dual PD Controller

Ours

RESULTS

Comparisons

Method		DIP-IMU							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	16.36	14.41	6.98	8.56	2.34	-	-	-
	TransPose [73]	13.97	7.62	4.90	5.83	0.13	0.85	0.59	-
Online	DIP [20]	17.10	15.16	7.33	8.96	3.01	-	-	117
	TransPose [73]	16.68	8.85	5.95	7.09	0.61	1.46	1.67	94
	PIP (Ours)	**15.02**	**8.73**	**5.04**	**5.95**	**0.23**	**0.24**	**0.12**	**16**

Method		TotalCapture							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	18.47	17.54	9.47	11.19	2.91	-	-	-
	TransPose [73]	14.71	12.19	5.44	6.22	0.16	0.91	0.76	-
Online	DIP [20]	18.62	17.22	9.42	11.22	3.62	-	-	117
	TransPose [73]	16.58	12.89	6.55	7.42	0.95	1.87	1.40	94
	PIP (Ours)	**12.93**	**12.04**	**5.61**	**6.51**	**0.20**	**0.20**	**0.23**	**16**

Better accuracy and physical correctness than all online methods

Comparable results with the best offline method

Compared with SOTA :

- pose error reduces 14%
- jitter reduces 87%
- latency reduces 81%

Comparisons

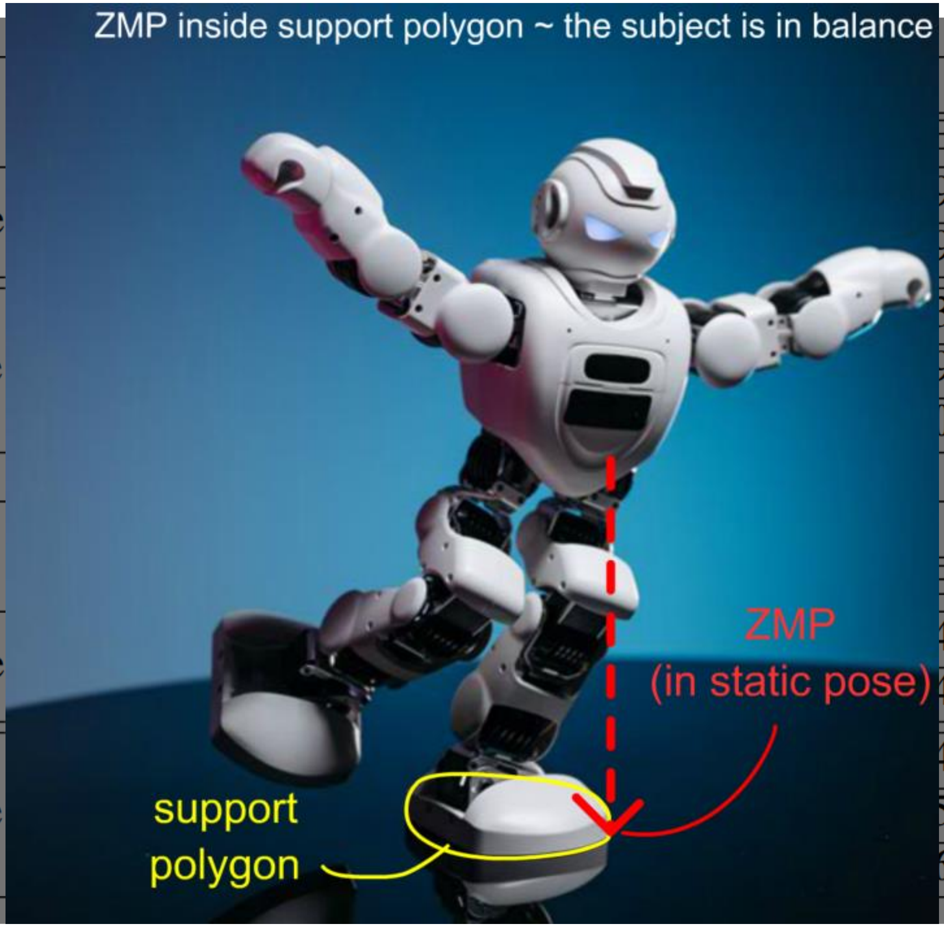
Method		DIP-IMU							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	16.36	14.41	6.98	8.56	2.34	-	-	-
	TransPose [73]	13.97	7.62	4.90	5.83	0.13	0.85	0.59	-
Online	DIP [20]	17.10	15.16	7.33	8.96	3.01	-	-	117
	TransPose [73]	16.68	8.85	5.95	7.09	0.61	1.46	1.67	94
	PIP (Ours)	**15.02**	**8.73**	**5.04**	**5.95**	**0.23**	**0.24**	**0.12**	**16**

Method		TotalCapture							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	18.47	17.54	9.47	11.19	2.91	-	-	-
	TransPose [73]	14.71	12.19	5.44	6.22	0.16	0.91	0.76	-
Online	DIP [20]	18.62	17.22	9.42	11.22	3.62	-	-	117
	TransPose [73]	16.58	12.89	6.55	7.42	0.95	1.87	1.40	94
	PIP (Ours)	**12.93**	**12.04**	**5.61**	**6.51**	**0.20**	**0.20**	**0.23**	**16**

Metrics for
pose accuracy

- SIP Error (degrees)： global orientation error of the upper arms and legs
- Angular Error (degrees): global orientation error of all joints
- Positional error (cm): joint position error (root aligned)
- Mesh error (cm): mesh vertex error (root aligned)

Comparisons



	DIP-IMU					
	Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	98	8.56	2.34	-	-	-
	90	5.83	0.13	0.85	0.59	-
Online	33	8.96	3.01	-	-	117
	95	7.09	0.61	1.46	1.67	94
	04	**5.95**	**0.23**	**0.24**	**0.12**	**16**

	TotalCapture					
	Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	47	11.19	2.91	-	-	-
	44	6.22	0.16	0.91	0.76	-
Online	42	11.22	3.62	-	-	117
	55	7.42	0.95	1.87	1.40	94
	61	**6.51**	**0.20**	**0.20**	**0.23**	**16**

Metrics for
physical correctness

- Relative jitter (km/s^3): jerk (time derivative of acceleration) of all joints (root fixed)
- Absolute jitter (km/s^3): jerk of all joints in the global space
- ZMP distance (m): Distance between zero-moment point (ZMP) and the support polygon (lower value for better equilibrium)

Comparisons

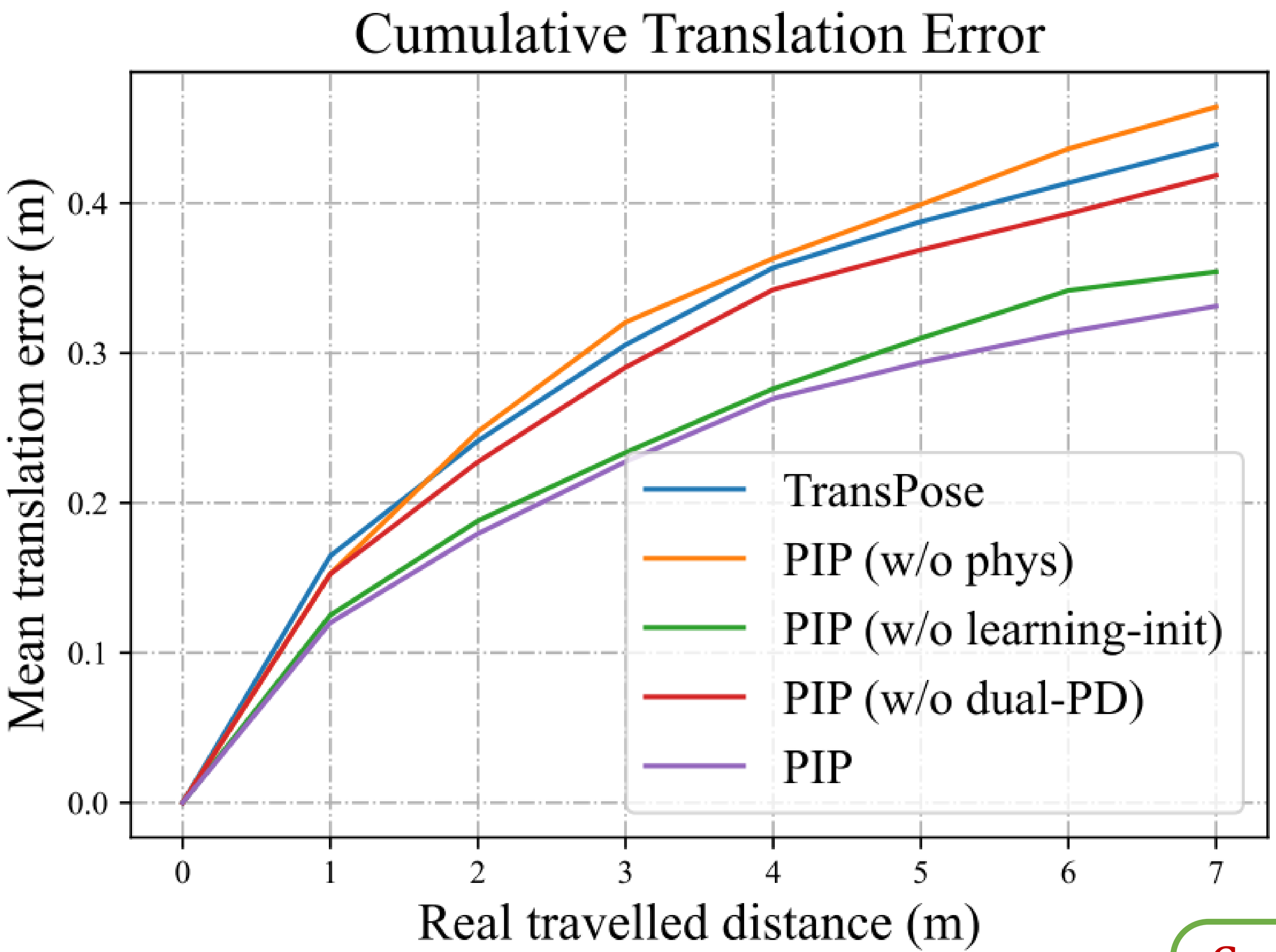
Method		DIP-IMU							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	16.36	14.41	6.98	8.56	2.34	-	-	-
	TransPose [73]	13.97	7.62	4.90	5.83	0.13	0.85	0.59	-
Online	DIP [20]	17.10	15.16	7.33	8.96	3.01	-	-	117
	TransPose [73]	16.68	8.85	5.95	7.09	0.61	1.46	1.67	94
	PIP (Ours)	**15.02**	**8.73**	**5.04**	**5.95**	**0.23**	**0.24**	**0.12**	**16**

Method		TotalCapture							
		SIP Error	Ang Error	Pos Error	Mesh Error	Rel Jitter	Abs Jitter	ZMP Dist	Latency
Offline	DIP [20]	18.47	17.54	9.47	11.19	2.91	-	-	-
	TransPose [73]	14.71	12.19	5.44	6.22	0.16	0.91	0.76	-
Online	DIP [20]	18.62	17.22	9.42	11.22	3.62	-	-	117
	TransPose [73]	16.58	12.89	6.55	7.42	0.95	1.87	1.40	94
	PIP (Ours)	**12.93**	**12.04**	**5.61**	**6.51**	**0.20**	**0.20**	**0.23**	**16**

Metrics for
real-time performance

- Latency (ms): time from receiving the inertia measurements to outputting the pose and translation for the corresponding frame

Comparisons

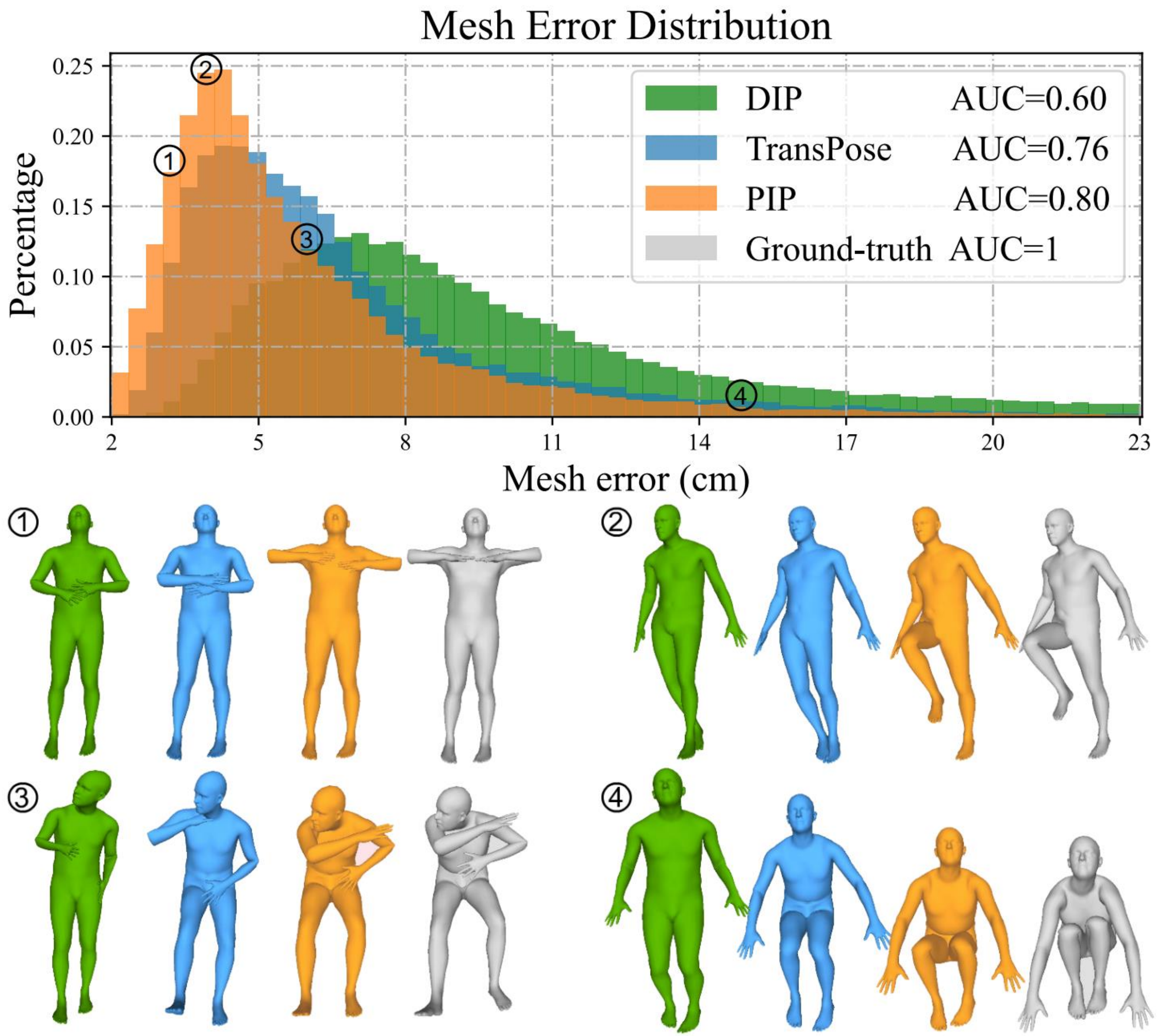


PIP Translation drift: 4.6%

Compared with SOTA :

- translation error reduces 26%

Comparisons



Examples picked at

1. 10%
2. mode
3. median
4. 95%

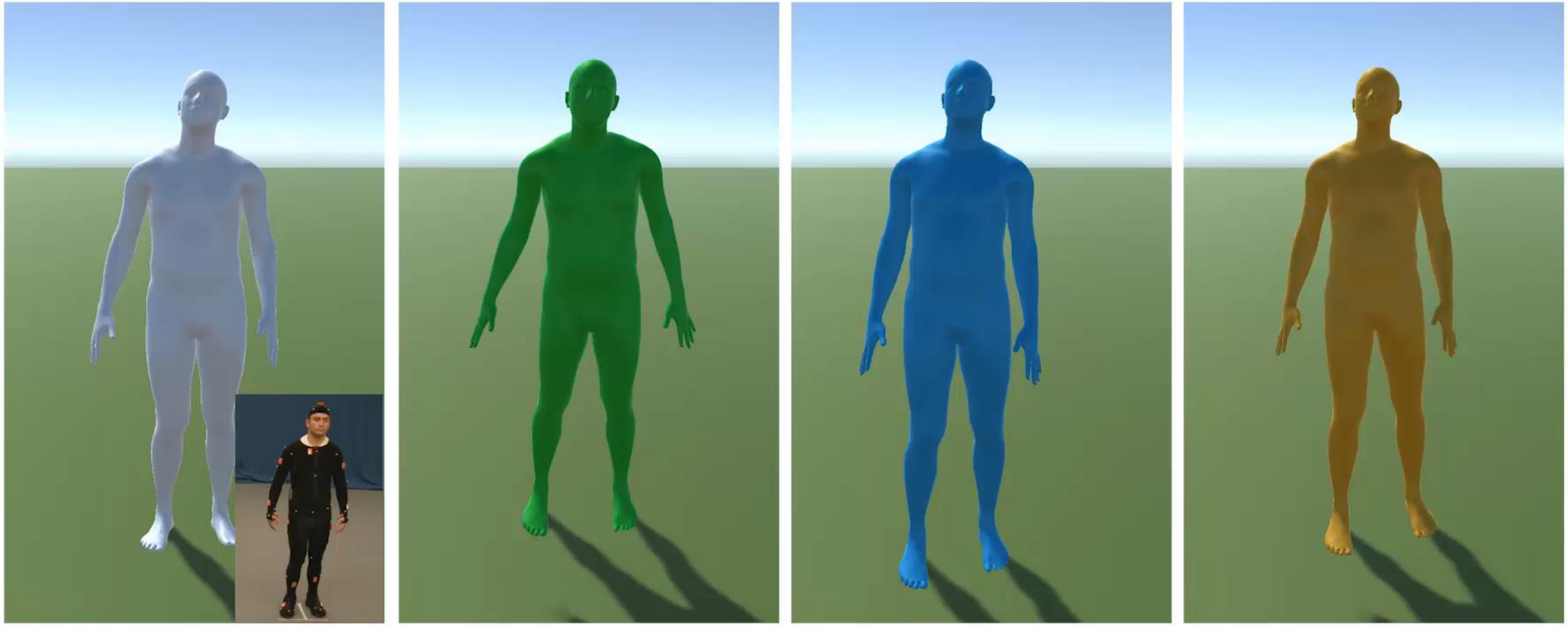
point from the mesh error distribution of TotalCapture dataset

Compared with SOTA :

- +4% AUC

Comparisons

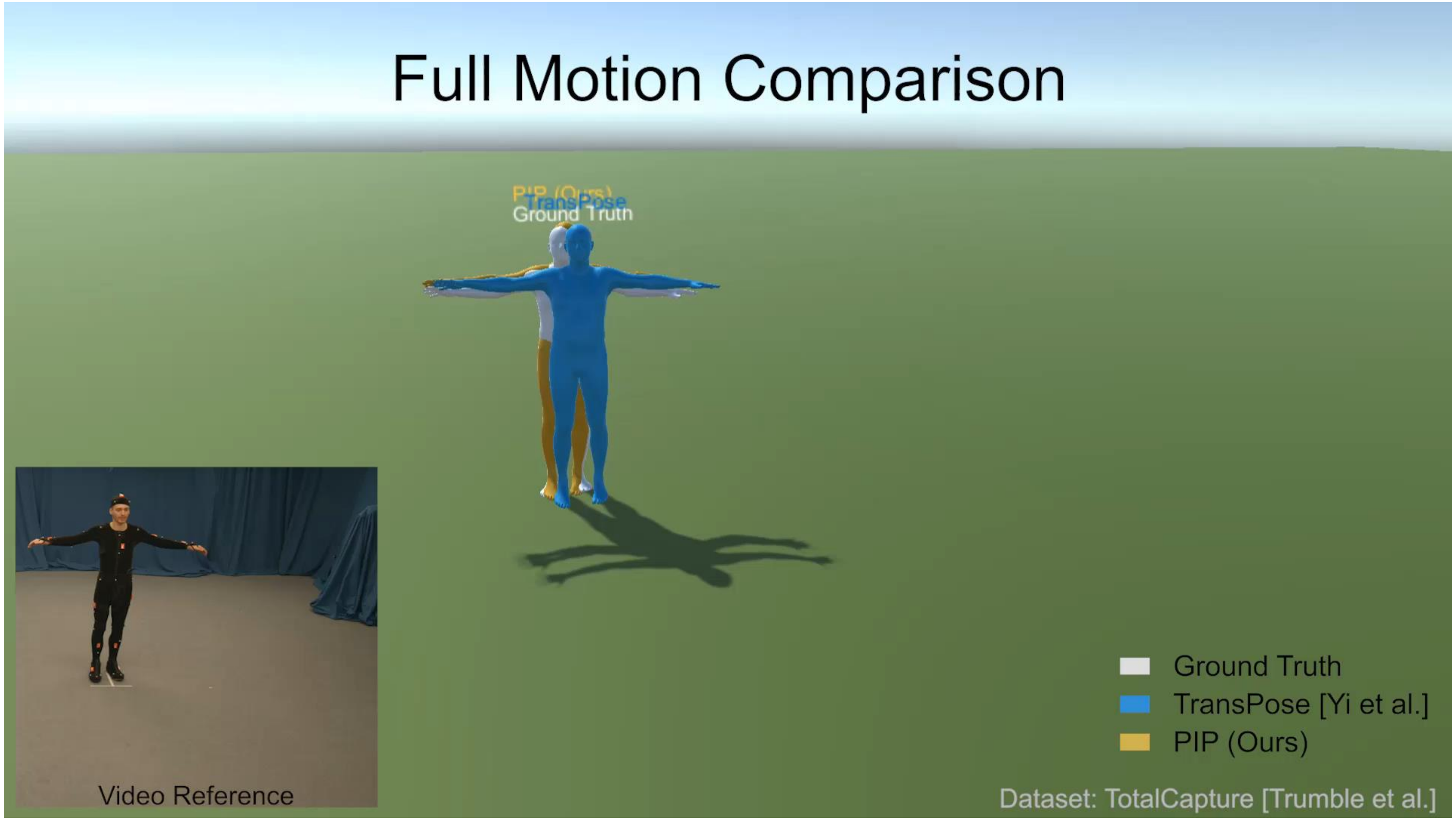
Pose Comparison



Ground Truth (with video reference) | DIP [Huang et al.] (no global position) | TransPose [Yi et al.] | PIP (Ours)

Dataset: TotalCapture [Trumble et al.]

Comparisons



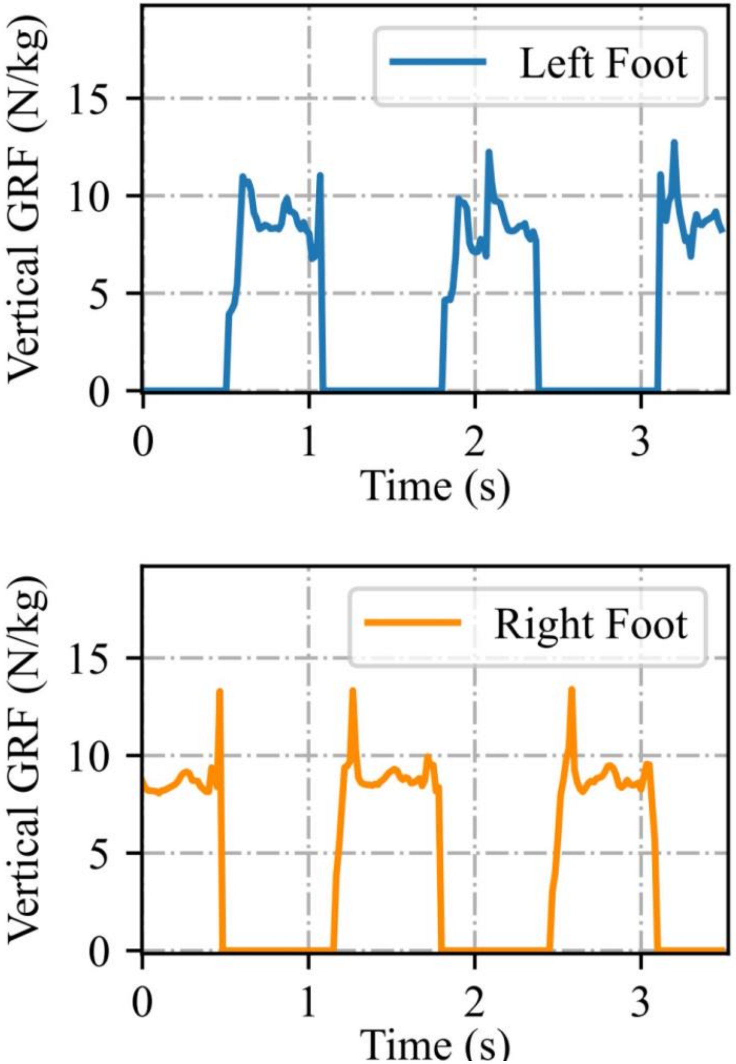
We ensure physical correctness by incorporating physics using a novel dual PD controller

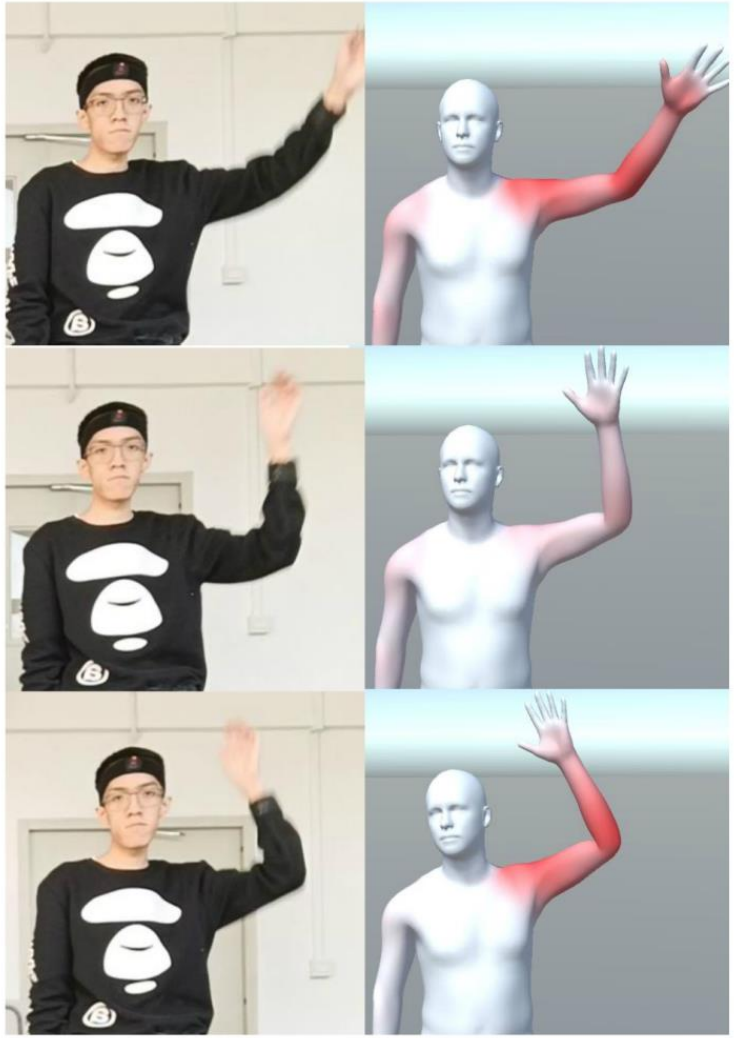
Evaluations

Method	DIP-IMU		TotalCapture	
	SIP Error	Jitter	SIP Error	Jitter
w/o learning-init	15.12	0.27	13.70	0.23
w/o dual-PD	15.04	0.28	12.93	0.32
w/o physics module	15.04	0.48	**12.84**	0.51
Ours	**15.02**	**0.24**	12.93	**0.20**

The physics module is helpful for estimating translation
and improving the physical correctness of the motion

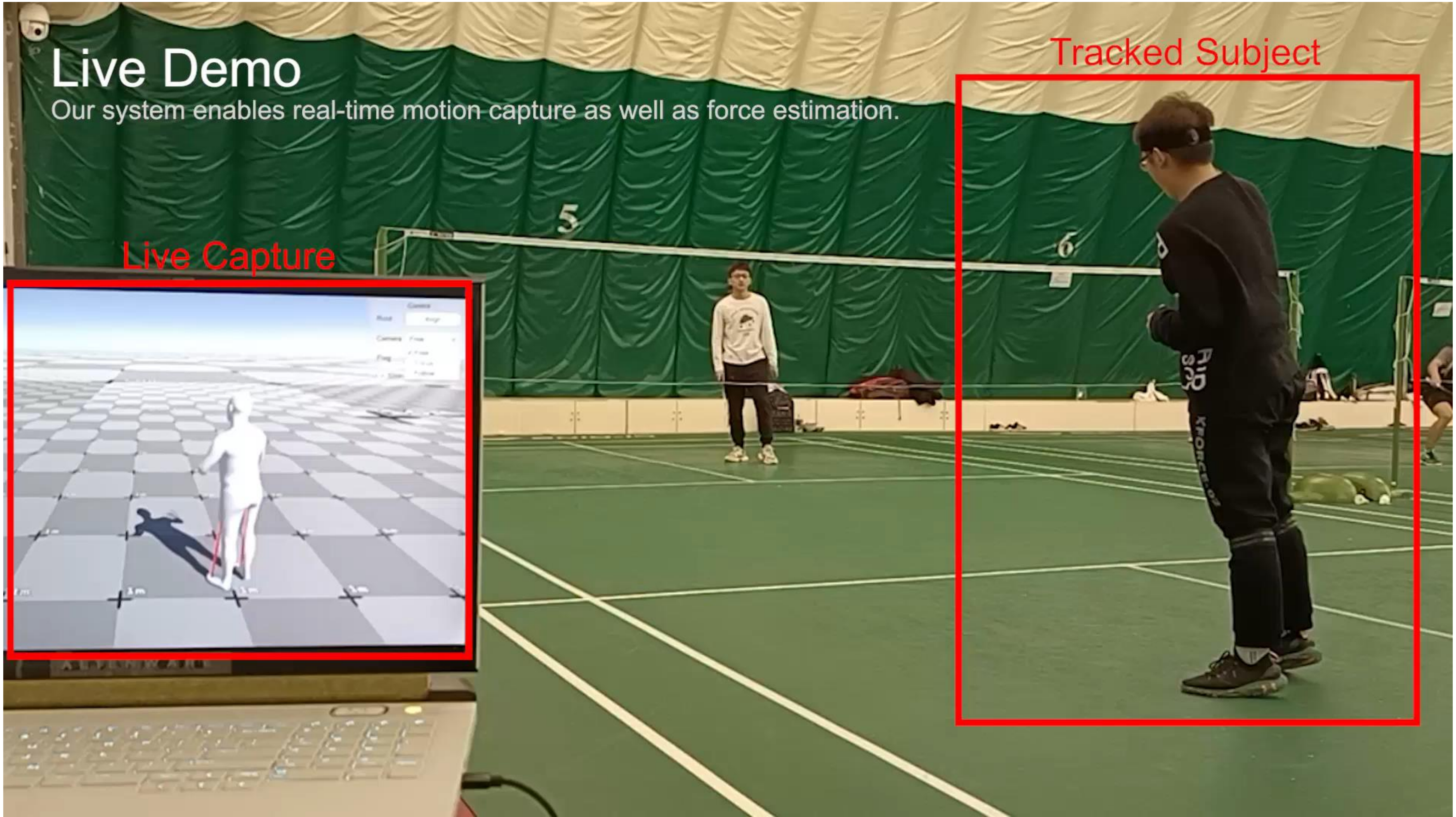
Evaluations





PIP generates plausible ground reaction forces and joint torques

In-the-wild Test



Limitations

Thank You!