DeepCap: Monocular Human Performance Capture Using Weak Supervision

Paper Authors: Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, Christian Theobalt

Presenter: Ji Yang


Input Image
Overlay
3D Views
Input Image
Overlay
3D Views

Background and Problem Formulation

Human performance capture is a highly important computer vision problem with many applications in movie production and virtual/augmented reality.

Previous methods:

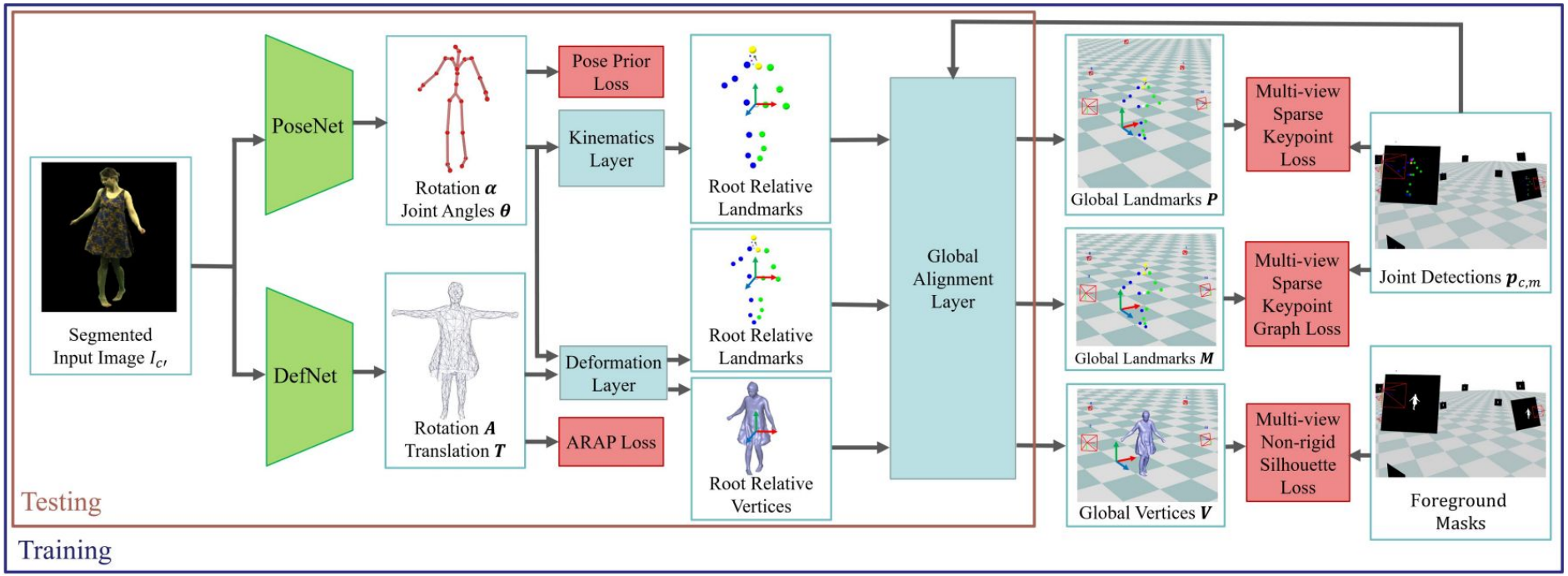
- multi-view marker-less methods rely on well-controlled multi-camera studios
- monocular human modeling approaches directly regress voxels or the continuous occupancy of the surface
- leverage deep learning-based sparse keypoint detections and perform an expensive template fitting afterwards, can only non-rigidly fit to the input view and suffer from instability

Background and Problem Formulation

Human performance capture is a highly important computer vision problem with many applications in movie production and virtual/augmented reality.

Proposed method:

- The first learning-based method that jointly infers the articulated and non-rigid 3D deformation parameters in a single feed-forward pass at much higher performance, accuracy and robustness.
- a CNN model which integrates a fully differentiable mesh template parameterized with pose and an embedded deformation graph


Segmented Input Image I_c
PoseNet
DefNet
Rotation α Joint Angles θ
Rotation A Translation T
Pose Prior Loss
Kinematics Layer
Deformation Layer
ARAP Loss
Root Relative Landmarks
Root Relative Landmarks
Root Relative Vertices
Global Alignment Layer
Global Landmarks P
Global Landmarks M
Global Vertices V
Multi-view Sparse Keypoint Loss
Multi-view Sparse Keypoint Graph Loss
Multi-view Non-rigid Silhouette Loss
Joint Detections $p_{c,m}$
Foreground Masks
Testing
Training

Method: Character Model

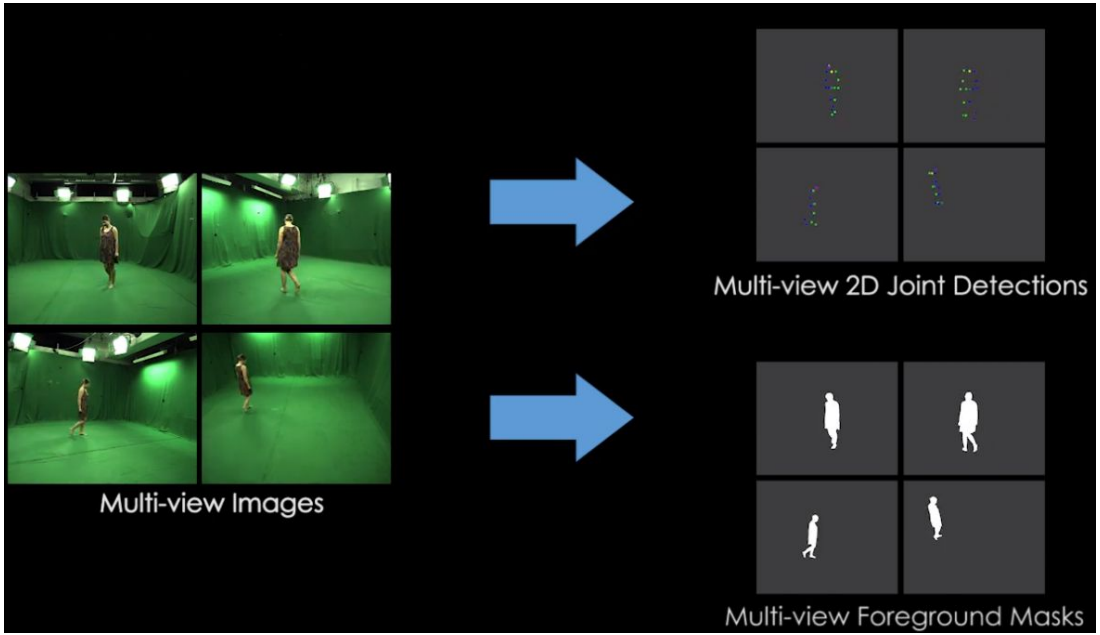
A person-specific 3D template model:

- Obtained by scanning the actor with a 3D scanner to obtain the textured mesh (Treedy's), rigged to a kinematic skeleton, which is parameterized with joint angles $\theta \in R^{27}$, the camera relative rotation $\alpha \in R^3$ and translation $t \in R^3$
- To model the non-rigid surface deformation, we automatically build an embedded deformation graph G with K nodes. The nodes are parameterized with Euler angles $A \in R^{K\times3}$ and translations $T \in R^{K\times3}$

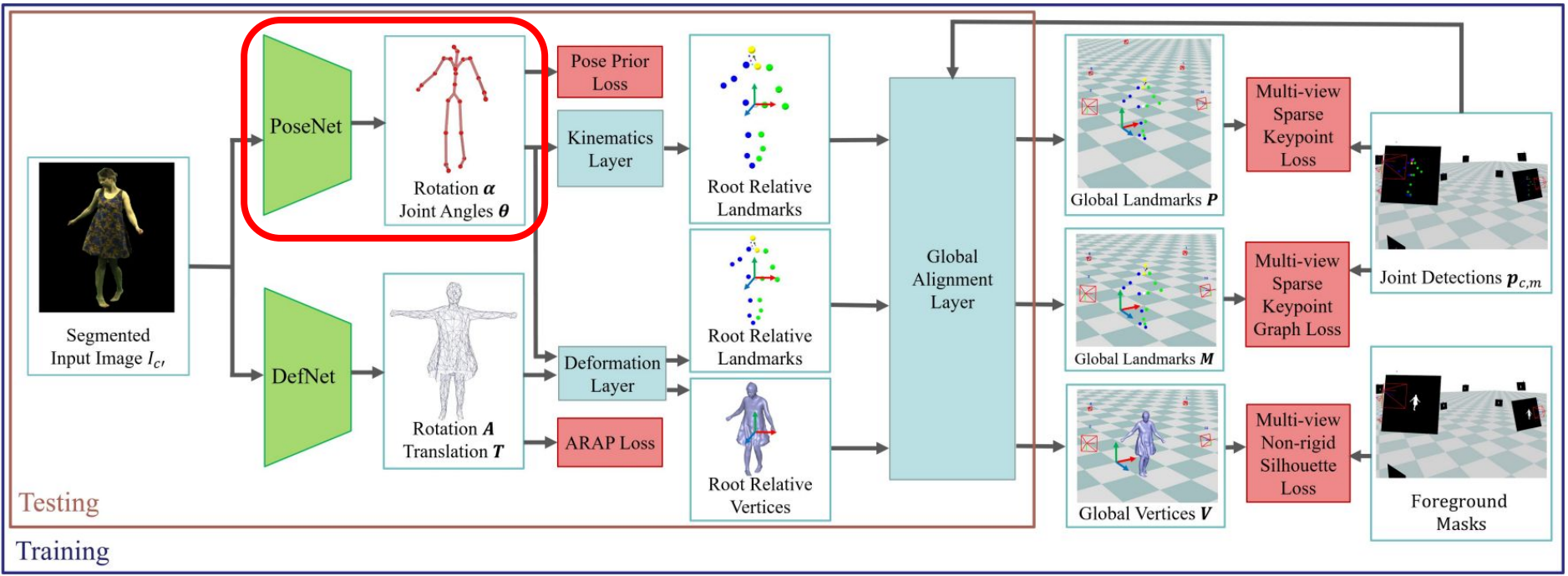
Method: Data

Recorded a multi-view video of the actor doing various actions in a calibrated multi-camera studio with green screen.

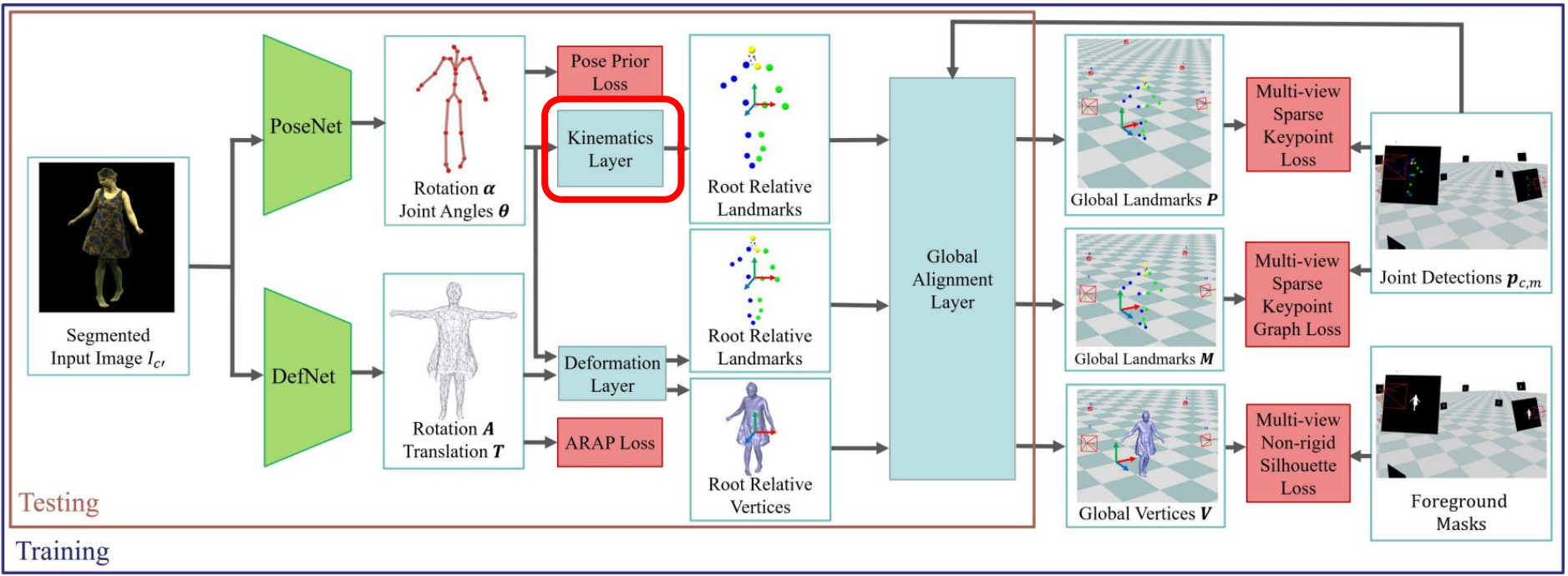
- 2D pose detection on the sequences using OpenPose and apply temporal filtering
- generate the foreground mask using color keying and compute the corresponding distance transform image D_f,c



Pose Network with ResNet50 Backbone



Pose Network: Kinematics Layer



Pose Network: Kinematics Layer

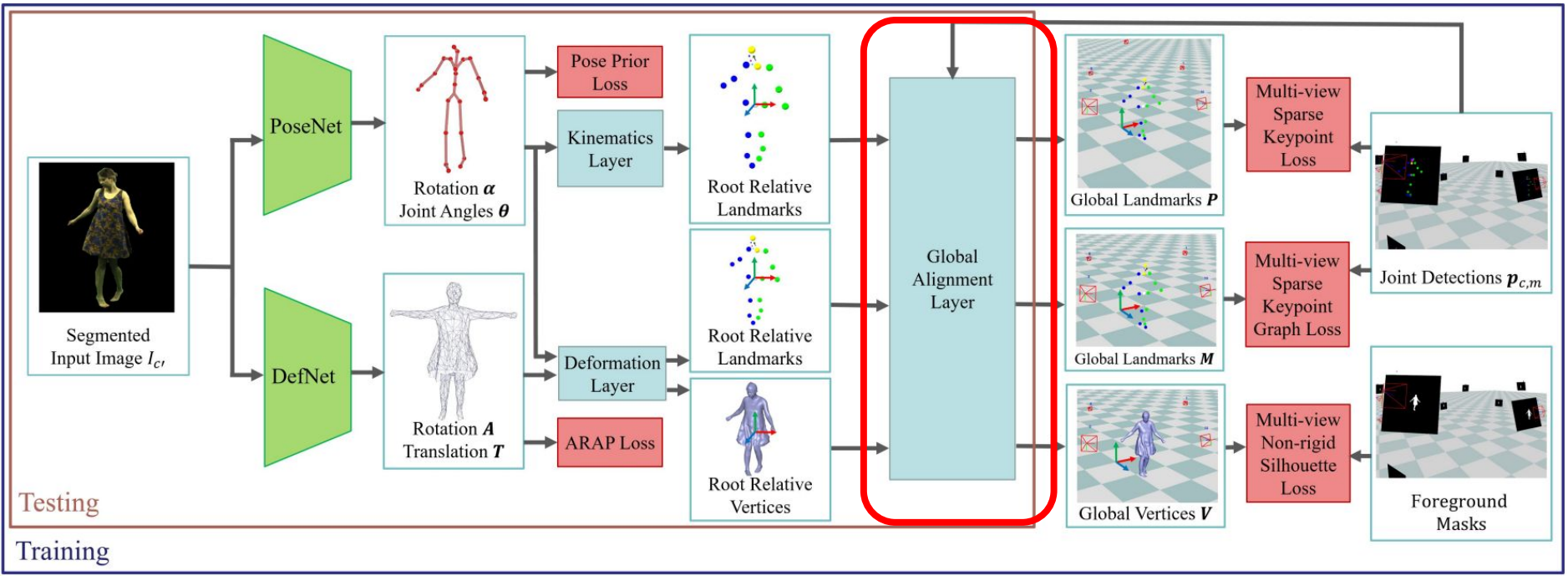
A kinematics layer as the differentiable function that takes the joint angles θ and the camera relative rotation α and computes the positions $\mathbf{P}_{c'} \in \mathbb{R}^{M\times 3}$ of the M 3D landmarks attached to the skeleton (17 body joints and 4 face landmarks).

The computed positions are in a camera-root-relative coordinate system, we can use

$$\mathbf{P}_m = \mathbf{R}_{c'}^T \mathbf{P}_{c',m} + \mathbf{t}$$

to transform $\mathbf{P}_{c'}$ to the world coordinate system.

Pose Network: Global Alignment Layer



Pose Network: Global Alignment Layer

It localizes the skeleton model in the world space, such that the globally rotated landmarks $\mathbf{R}_{c'}^T \mathbf{P}_{c',m}$ project onto the corresponding detections in all camera views.

This is done by minimizing $$\sum_c \sum_m \sigma_{c,m} \| (\mathbf{R}_{c'}^T \mathbf{P}_{c',m} + \mathbf{t} - \mathbf{o}_c) \times \mathbf{d}_{c,m} \|^2$$

And $\mathbf{d}_{c,m}$ is obtained from $$\mathbf{d}_{c,m} = \frac{(\mathbf{E}_c^{-1} \tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c}{\| (\mathbf{E}_c^{-1} \tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c \|}.$$

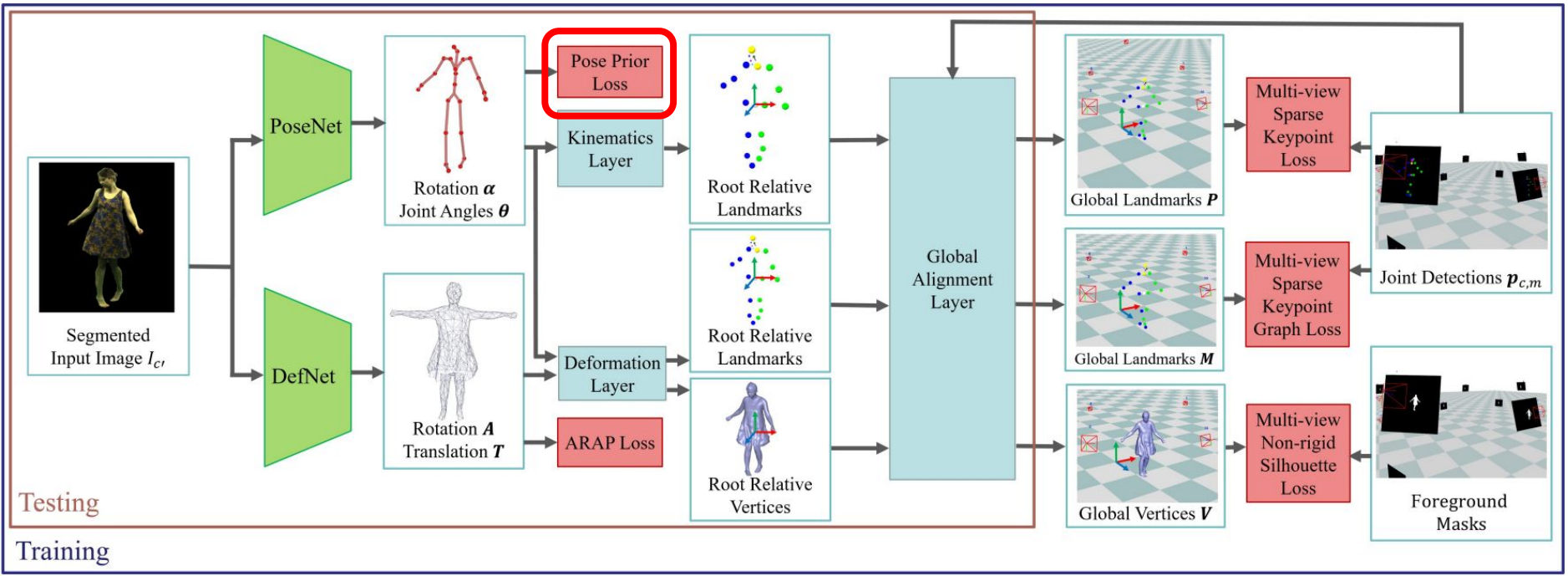
The minimization problem of Eq. 2 can be solved in closed form:

$$\mathbf{t} = \mathbf{W}^{-1} \sum_{c,m} \mathbf{D}_{c,m} (\mathbf{R}_{c'}^T \mathbf{P}_{c',m} - \mathbf{o}_c) + \mathbf{o}_c - \mathbf{R}_{c'}^T \mathbf{P}_{c',m},$$

where

$$\mathbf{W} = \sum_c \sum_m \mathbf{I} - \mathbf{D}_{c,m}.$$

Losses: Pose Prior Loss



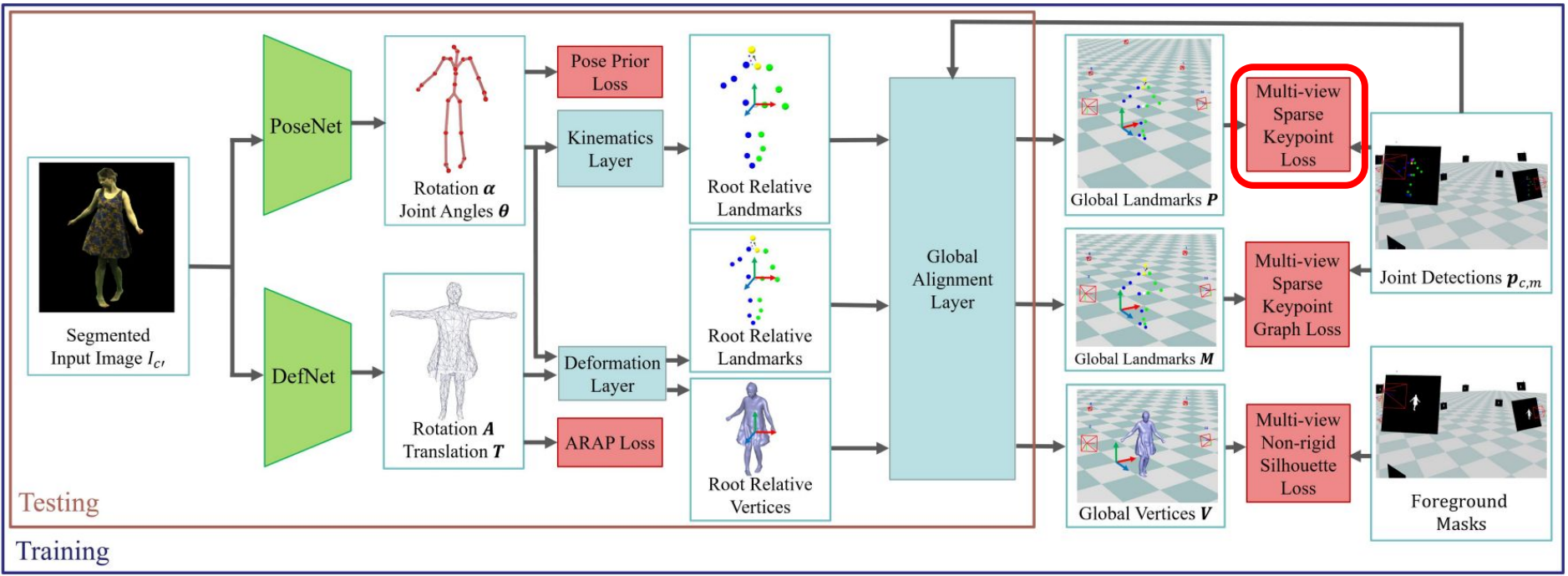
Losses: Pose Prior Loss

To avoid unnatural poses, the authors impose a pose prior loss on the joint angles

$$\mathcal{L}_{\text{limit}}(\boldsymbol{\theta}) = \sum_{i=1}^{27} \Psi(\boldsymbol{\theta}_i)$$

$$\Psi(x) = \begin{cases} (x - \boldsymbol{\theta}_{\max,i})^2, & \text{if } x > \boldsymbol{\theta}_{\max,i} \\ (\boldsymbol{\theta}_{\min,i} - x)^2, & \text{if } x < \boldsymbol{\theta}_{\min,i} \\ 0, & \text{otherwise} \end{cases}$$

Losses: Sparse Keypoint Loss

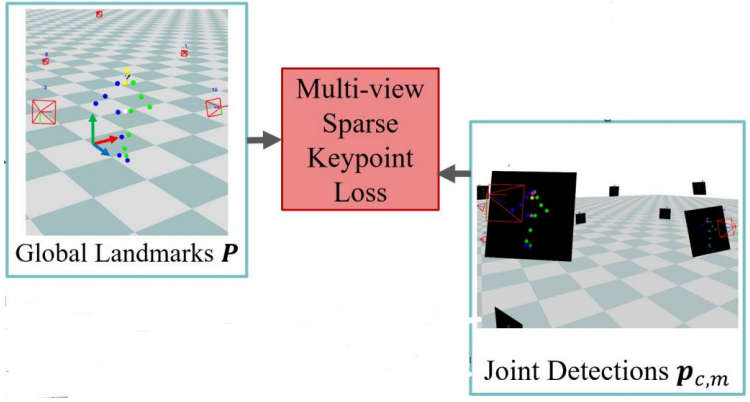


Losses: Sparse Keypoint Loss

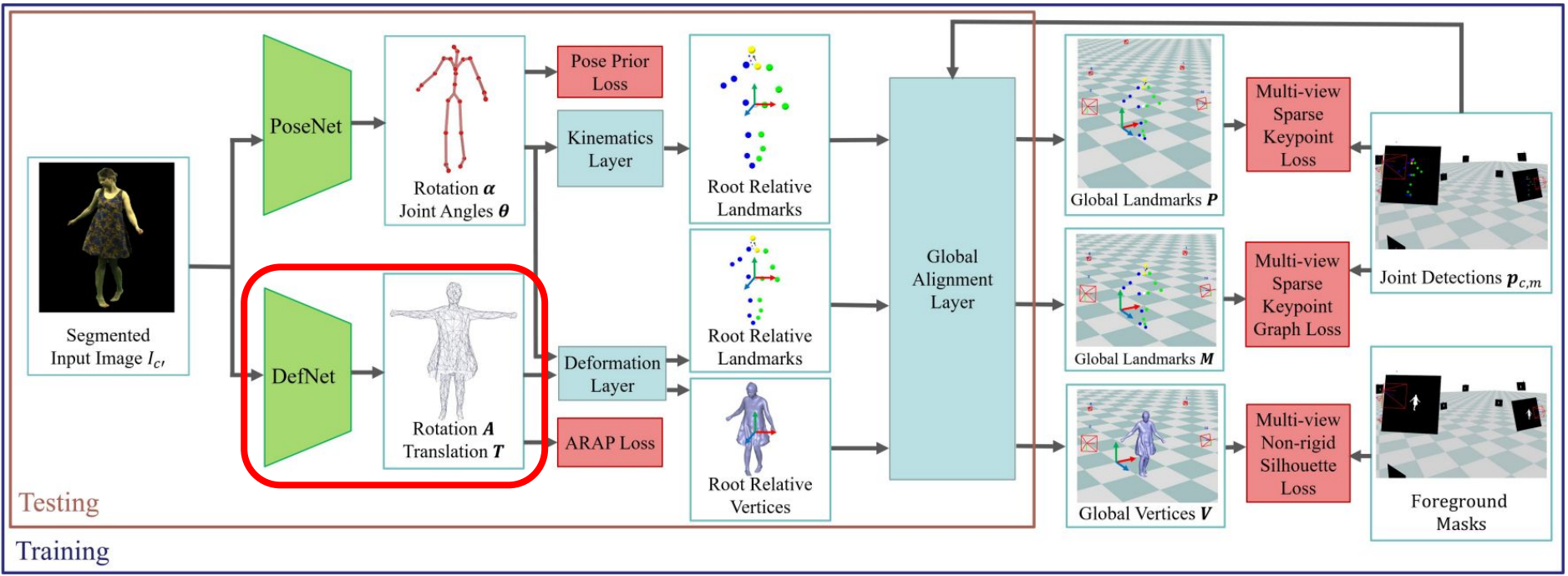
2D sparse keypoint loss for the PoseNet can be expressed as

$$\mathcal{L}_{\mathrm{kp}}(\mathbf{P}) = \sum_{c} \sum_{m} \lambda_m \sigma_{c,m} \left\| \pi_c \left(\mathbf{P}_m \right) - \mathbf{p}_{c,m} \right\|^2$$

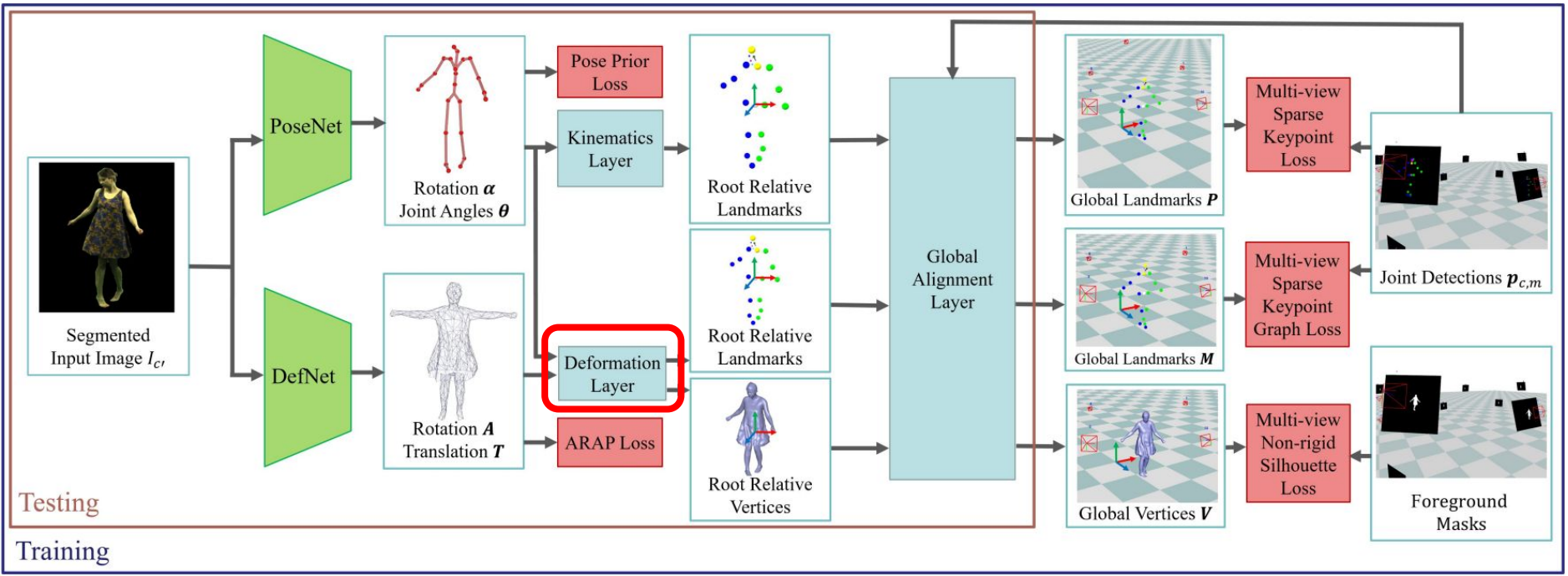
which ensures that each landmark projects onto the corresponding 2D joint detections $\mathbf{p}_{c,m}$ in all camera views



Deformation Network



Deformation Network: Deformation Layer



Deformation Network: Deformation Layer

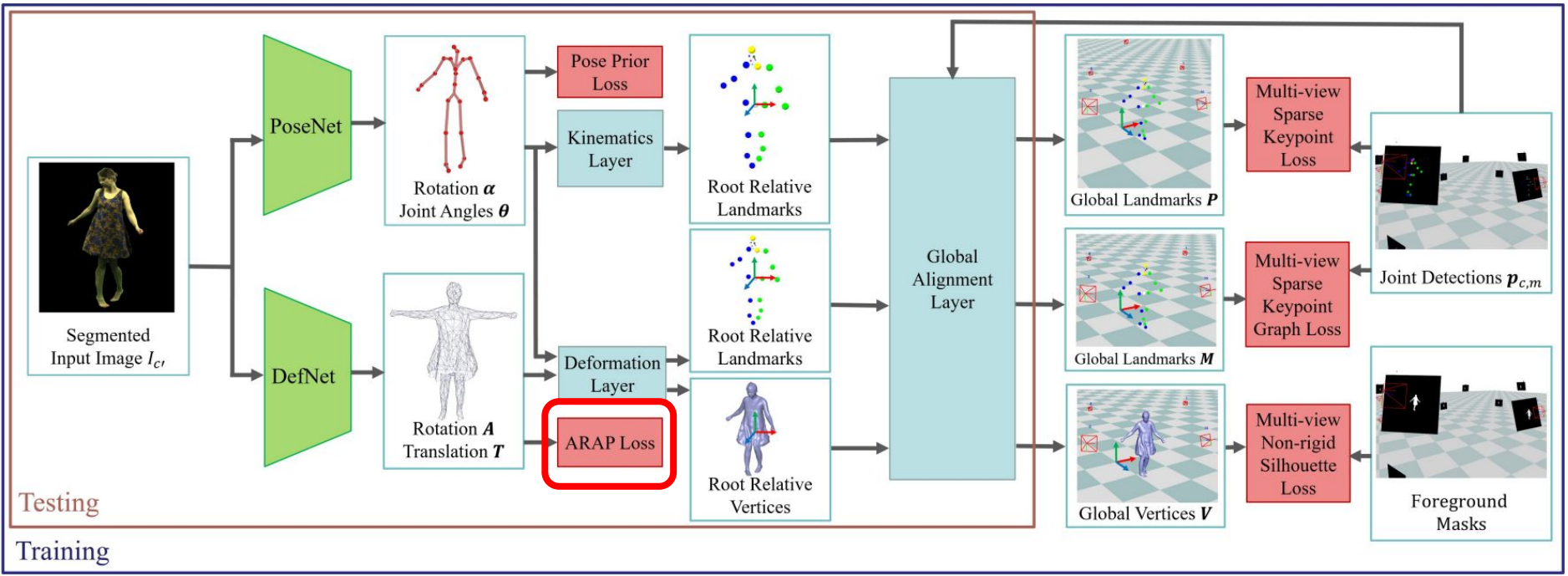
The deformation layer takes A and T from DefNet as input to non-rigidly deform the surface

$$\mathbf{Y}_i = \sum_{k \in \mathcal{N}_{\mathrm{vn}}(i)} w_{i,k}(R(\mathbf{A}_k)(\hat{\mathbf{V}}_i - \mathbf{G}_k) + \mathbf{G}_k + \mathbf{T}_k).$$

The skeletal pose is applied on the deformed mesh vertices to obtain the vertex positions in the input camera space

$$\mathbf{V}_{\mathrm{c}',i} = \sum_{k \in \mathcal{N}_{\mathrm{vn}}(i)} w_{i,k}(R_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})\mathbf{Y}_i + t_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha}))$$

Losses: As-rigid-as-possible Loss



Losses: As-rigid-as-possible Loss

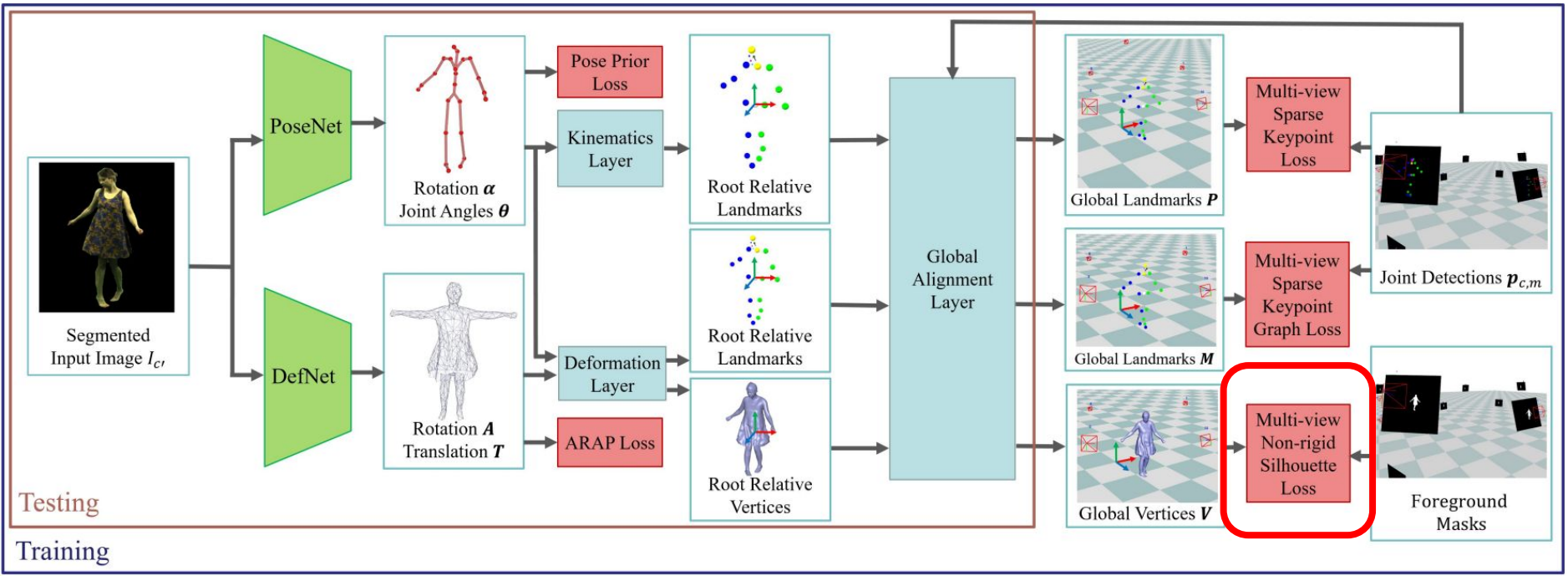
To enforce local smoothness of the surface, an as-rigid-as-possible loss is imposed:

$$\mathcal{L}_{\text{arap}}(\mathbf{A}, \mathbf{T}) = \sum_{k} \sum_{l \in \mathcal{N}_{\text{n}}(k)} u_{k,l} \| d_{k,l}(\mathbf{A}, \mathbf{T}) \|_1$$

where

$$d_{k,l}(\mathbf{A}, \mathbf{T}) = R(\mathbf{A}_k)(\mathbf{G}_l - \mathbf{G}_k) + \mathbf{T}_k + \mathbf{G}_k - (\mathbf{G}_l + \mathbf{T}_l)$$

Losses: Non-rigid Silhouette Loss

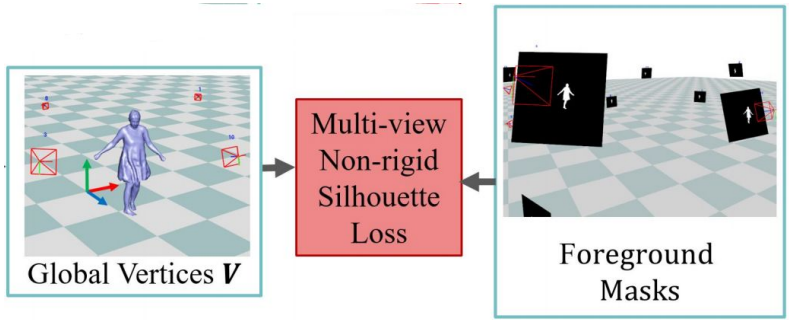


Losses: Non-rigid Silhouette Loss

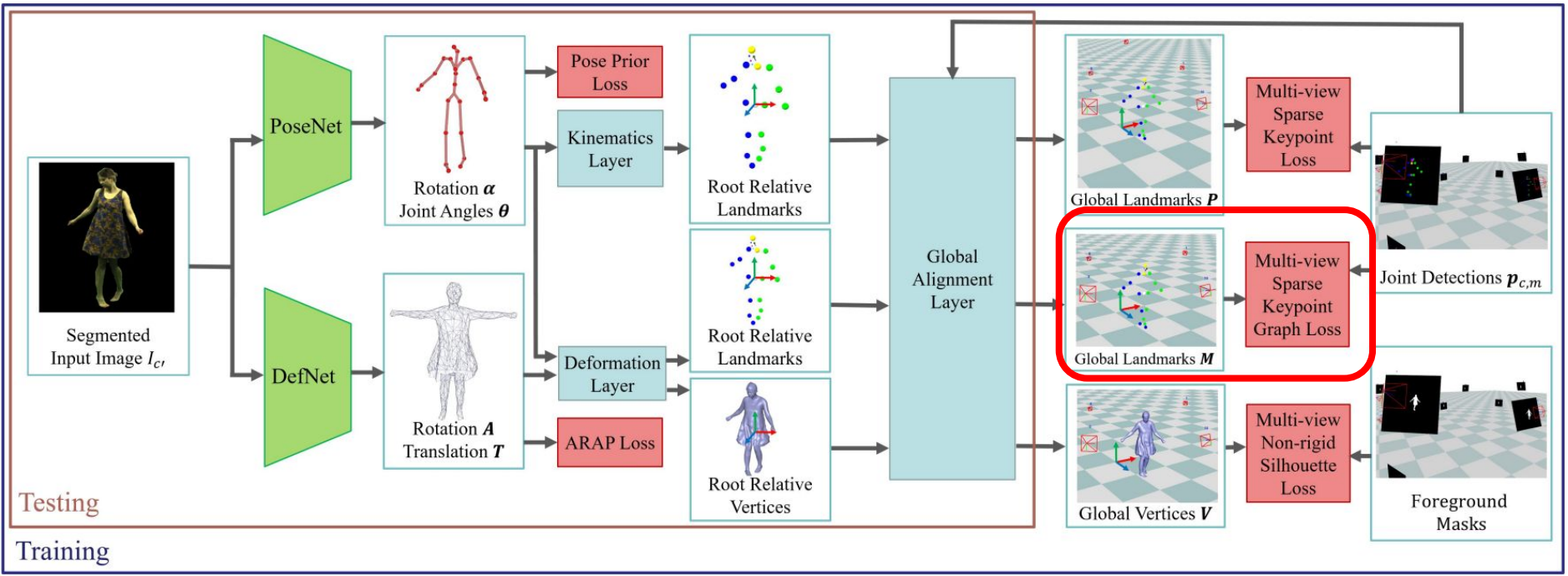
This loss encourages that the non-rigidly deformed mesh matches the multi-view silhouettes in all camera views. It can be formulated using the distance transform representation

$$\mathcal{L}_{\text{sil}}(\mathbf{V}) = \sum_{c} \sum_{i \in \mathcal{B}_c} \rho_{c,i} \| D_c \left(\pi_c \left(\mathbf{V}_i \right) \right) \|^2$$

The silhouette loss ensures that the boundary vertices project onto the zero-set of the distance transform, i.e., the foreground silhouette



Losses: Sparse Keypoint Graph Loss

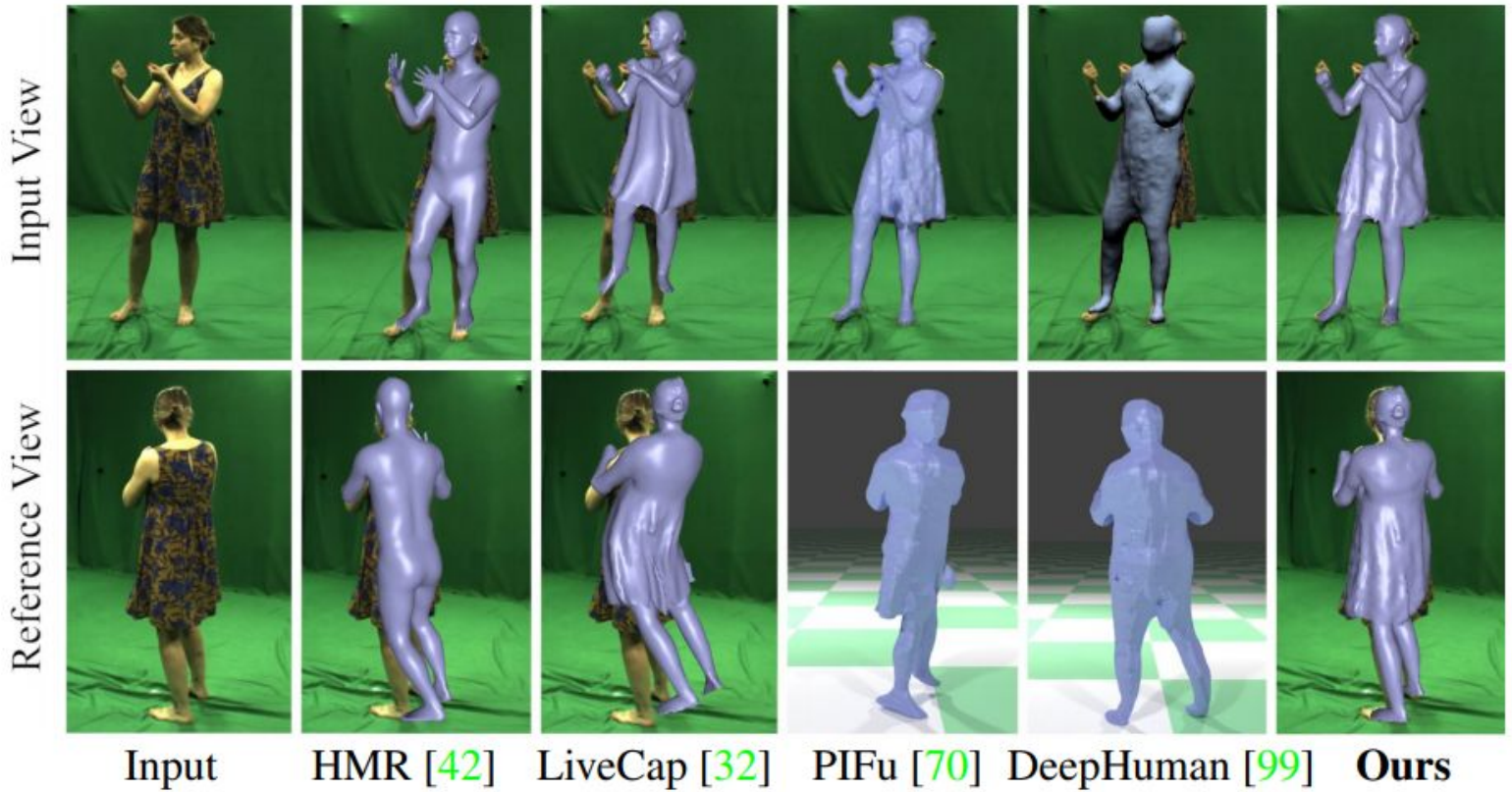


Losses: Sparse Keypoint Graph Loss

A sparse keypoint loss to constrain the mesh deformation, which is similar to the keypoint loss for PoseNet

$$\mathcal{L}_{\text{kpg}}(\mathbf{M}) = \sum_{c} \sum_{m} \sigma_{c,m} \| \pi_c (\mathbf{M}_m) - \mathbf{p}_{c,m} \|^2$$

Qualitative Results



Qualitative Results

Qualitative Results

Quantitative Results: Skeletal Pose Accuracy

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S1				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect [53]	-	66.06	28.02	77.19
HMR [42]	-	82.39	43.61	72.61
HMMR [43]	-	87.48	45.33	72.40
LiveCap [32]	317.01	71.13	37.90	92.84
Ours	**91.08**	**98.43**	**58.71**	**49.11**
MVBL	76.03	99.17	57.79	45.44

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S4				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect [53]	-	82.06	42.73	72.62
HMR [42]	-	86.88	43.91	73.63
HMMR [43]	-	82.80	41.18	77.41
LiveCap [32]	248.67	75.11	37.35	83.48
Ours	**96.56**	**96.74**	**59.25**	**45.40**
MVBL	75.82	96.20	57.27	45.12

consistently better than other monocular approaches

even close to the multi-view baseline

Quantitative Results: Surface Reconstruction Accuracy

AMVIoU, RVIoU, and SVIoU (in %) on S1 sequence			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR [42]	62.25	61.7	68.85
HMMR [43]	65.98	65.58	70.77
LiveCap [32]	56.02	54.21	77.75
DeepHuman [99]	-	-	**91.57**
Ours	**87.2**	**87.03**	89.26
MVBL	91.74	91.72	92.02

AMVIoU, RVIoU, and SVIoU (in %) on S4 sequence			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR [42]	65.1	64.66	70.84
HMMR [43]	63.79	63.29	70.23
LiveCap [32]	59.96	59.02	72.16
DeepHuman [99]	-	-	84.15
Ours	**82.53**	**82.22**	**86.66**
MVBL	88.14	88.03	89.66

Ablative Studies

3DPCK and AMVIoU (in %) on S4 sequence		
Method	**3DPCK↑**	**AMVIoU↑**
1 camera view	62.11	65.11
2 camera views	93.52	78.44
3 camera views	94.70	79.75
7 camera views	95.95	81.73
6500 frames	85.19	73.41
13000 frames	92.25	78.97
PoseNet-only	96.74	78.51
Ours(14 views, 26000 frames)	**96.74**	**82.53**

Ablative Studies