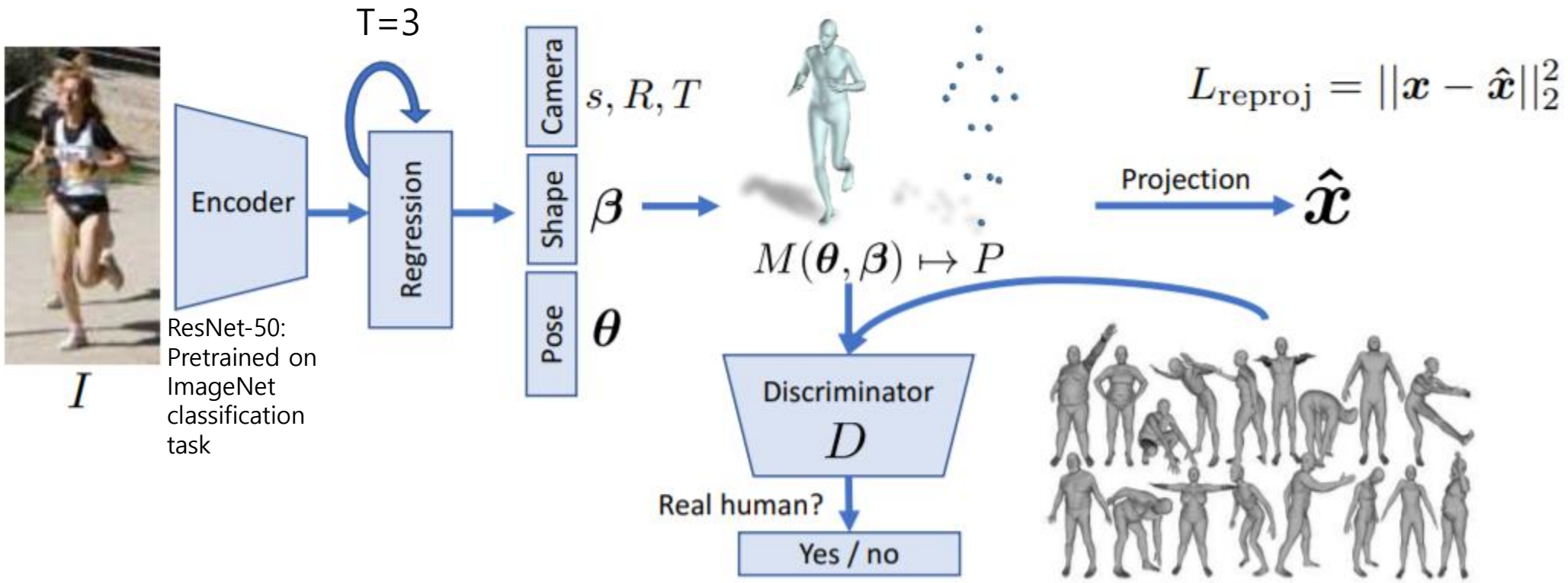
HMR

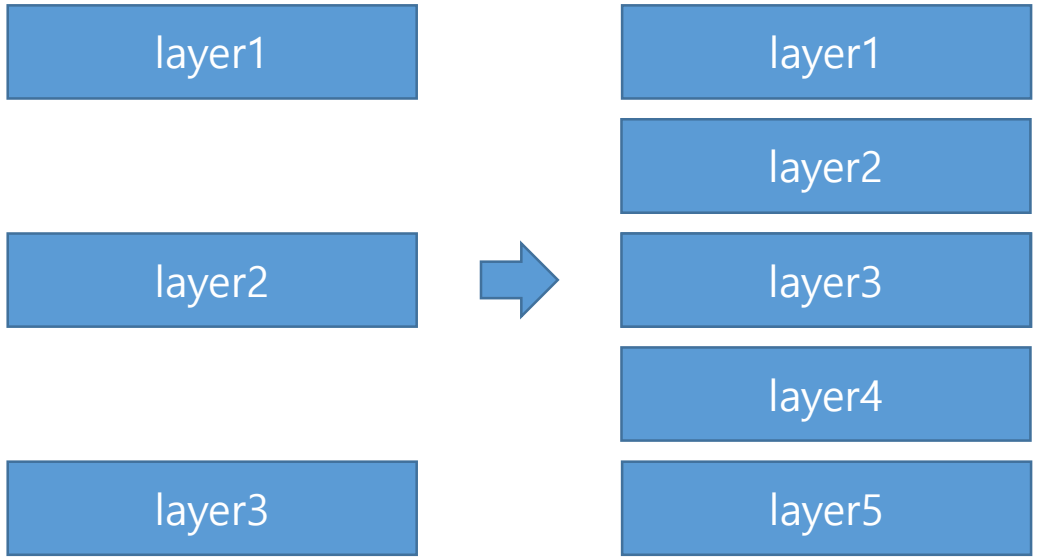
Human Mesh Recovery

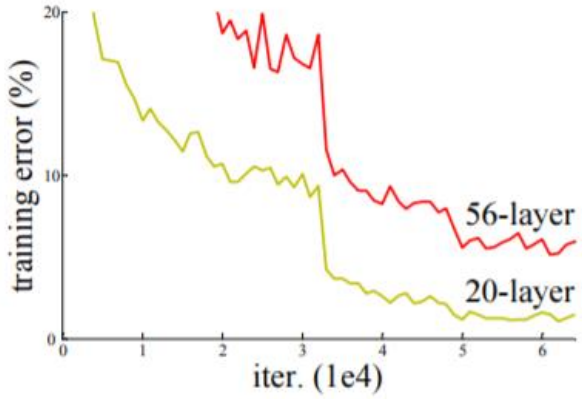
End-to-end Recovery of Human Shape and Pose

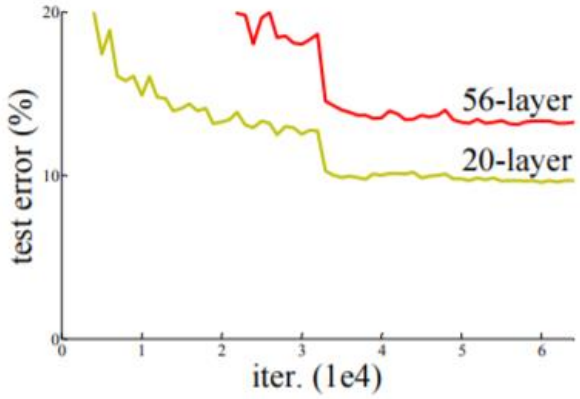
HMR


T=3
Encoder
Regression
Camera
Shape
Pose
s, R, T
β
θ
I
ResNet-50: Pretrained on ImageNet classification task
$M(\theta, \beta) \mapsto P$
Projection
$\hat{x}$
$L_{\text{reproj}} = ||x - \hat{x}||_2^2$
Discriminator
D
Real human?
Yes / no

HMR: Encoder: Resnet-50

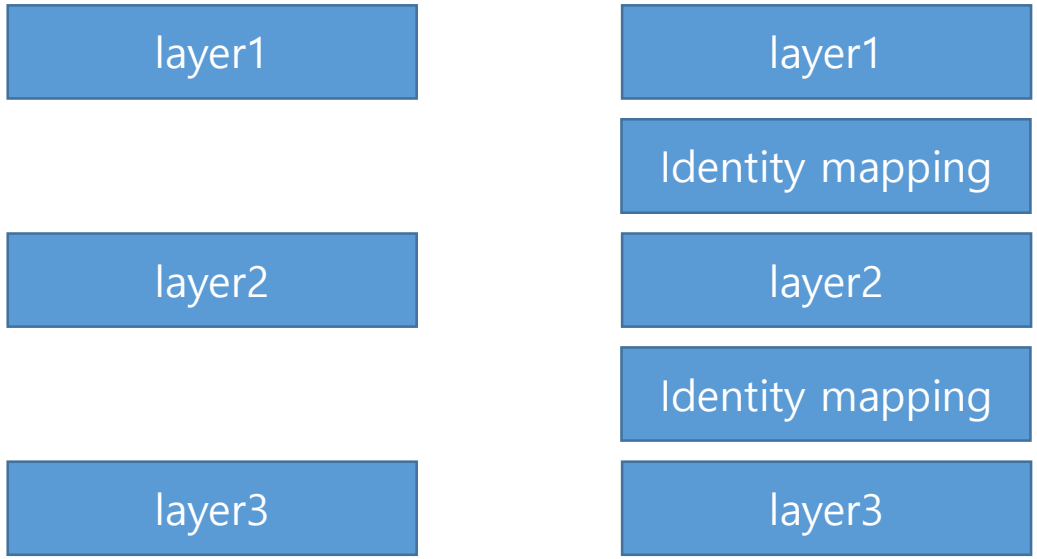






- When deeper networks start converging, a degradation problem has been exposed
- Unexpectedly, such degradation is not caused by overfitting
- The degradation indicates that not all systems are similarly easy to optimize. **(vanishing gradient problem)**

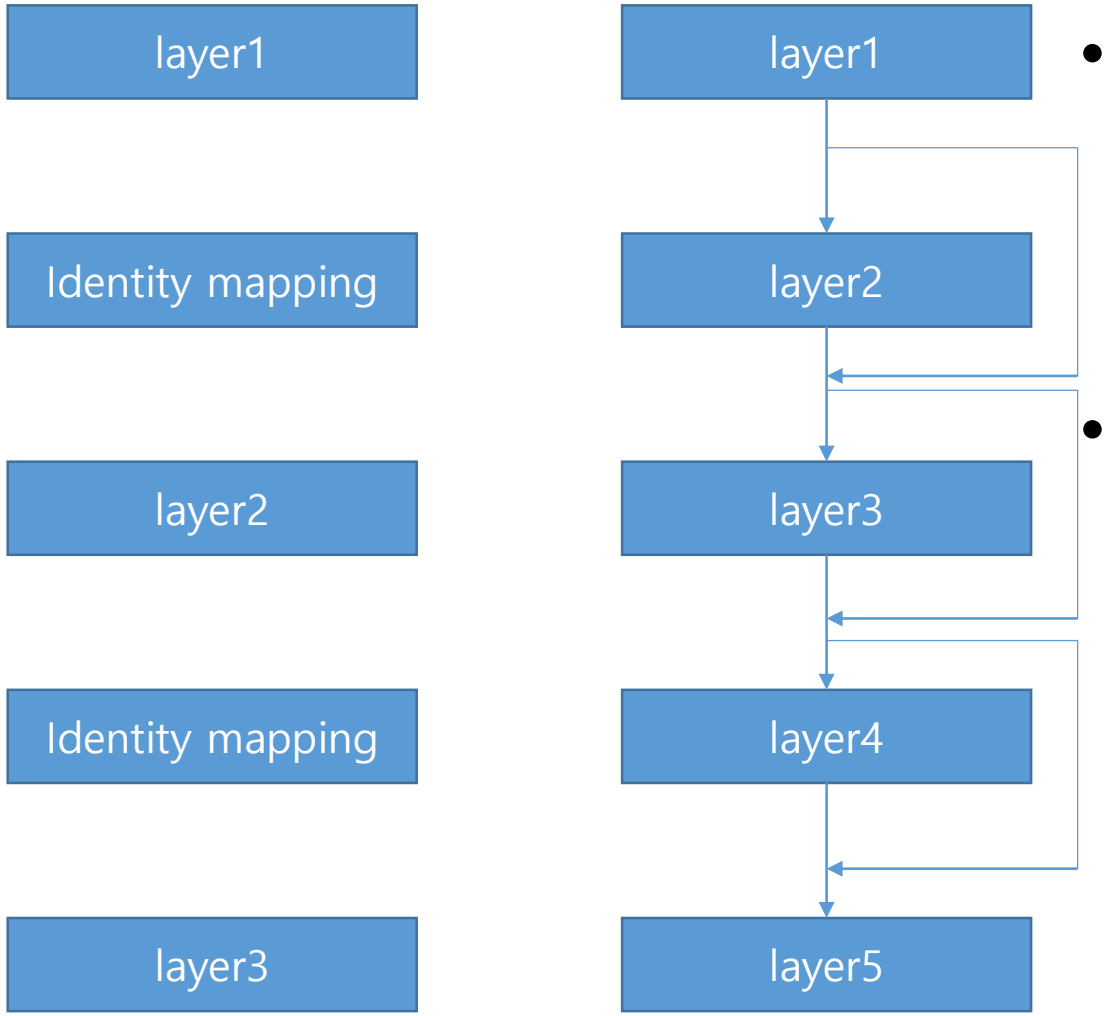
HMR: Encoder: Resnet-50



Constructed solution

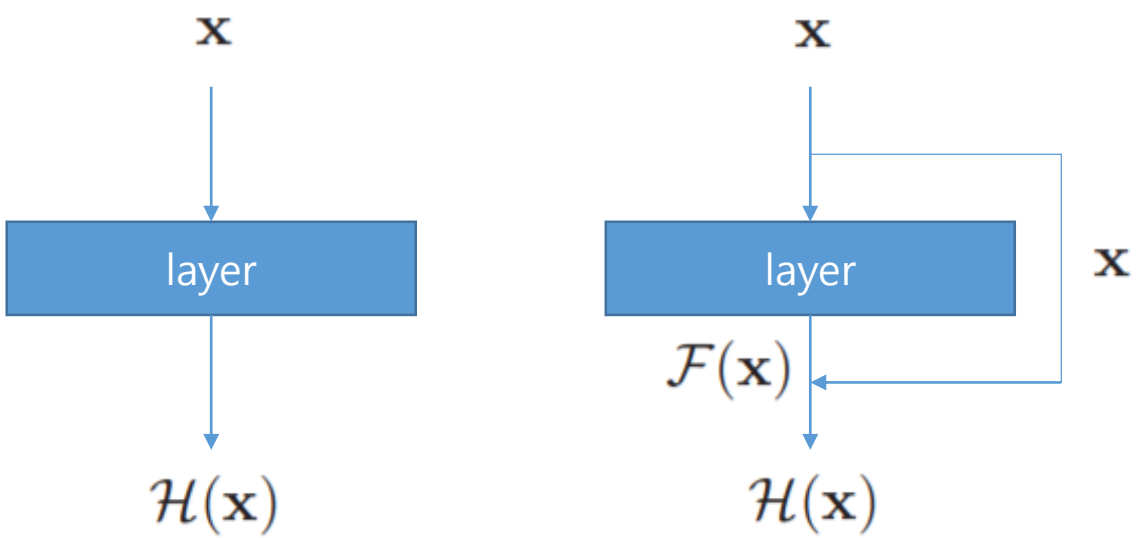
- The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart.
- But experiments show that our deeper model can’t find solutions that are better than the constructed solution

HMR: Encoder: Resnet-50



- the solvers might have difficulties in approximating identity mappings
- with the residual learning, if identity mappings are optimal, the solvers may simply drive the weights of layers toward zero.

HMR: Encoder: Resnet-50



$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}.$$

- Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping.
- In our case, the shortcut connections simply perform identity mapping

HMR: Encoder: Resnet-50

- We show by experiments that the learned residual functions in general have small responses, suggesting that identity mappings provide reasonable preconditioning.

Small responses

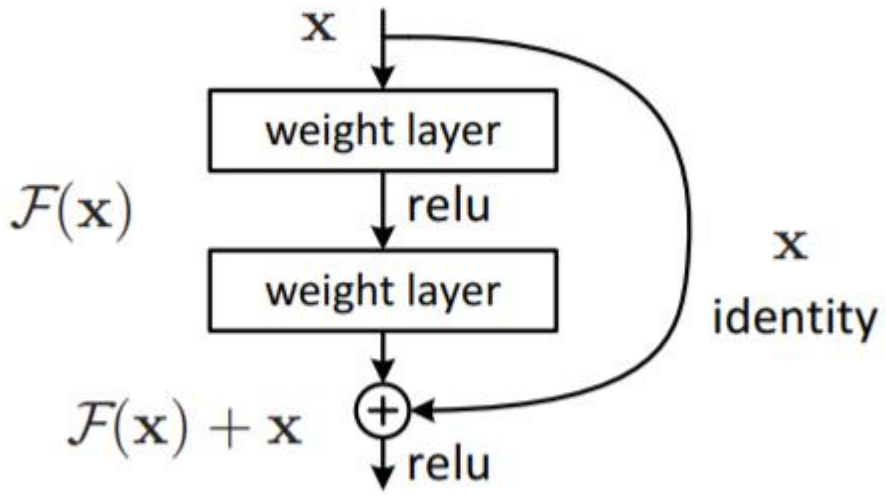


Figure 2. Residual learning: a building block.

HMR: Encoder: Resnet-50

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

- The dimensions of x and F must be equal.

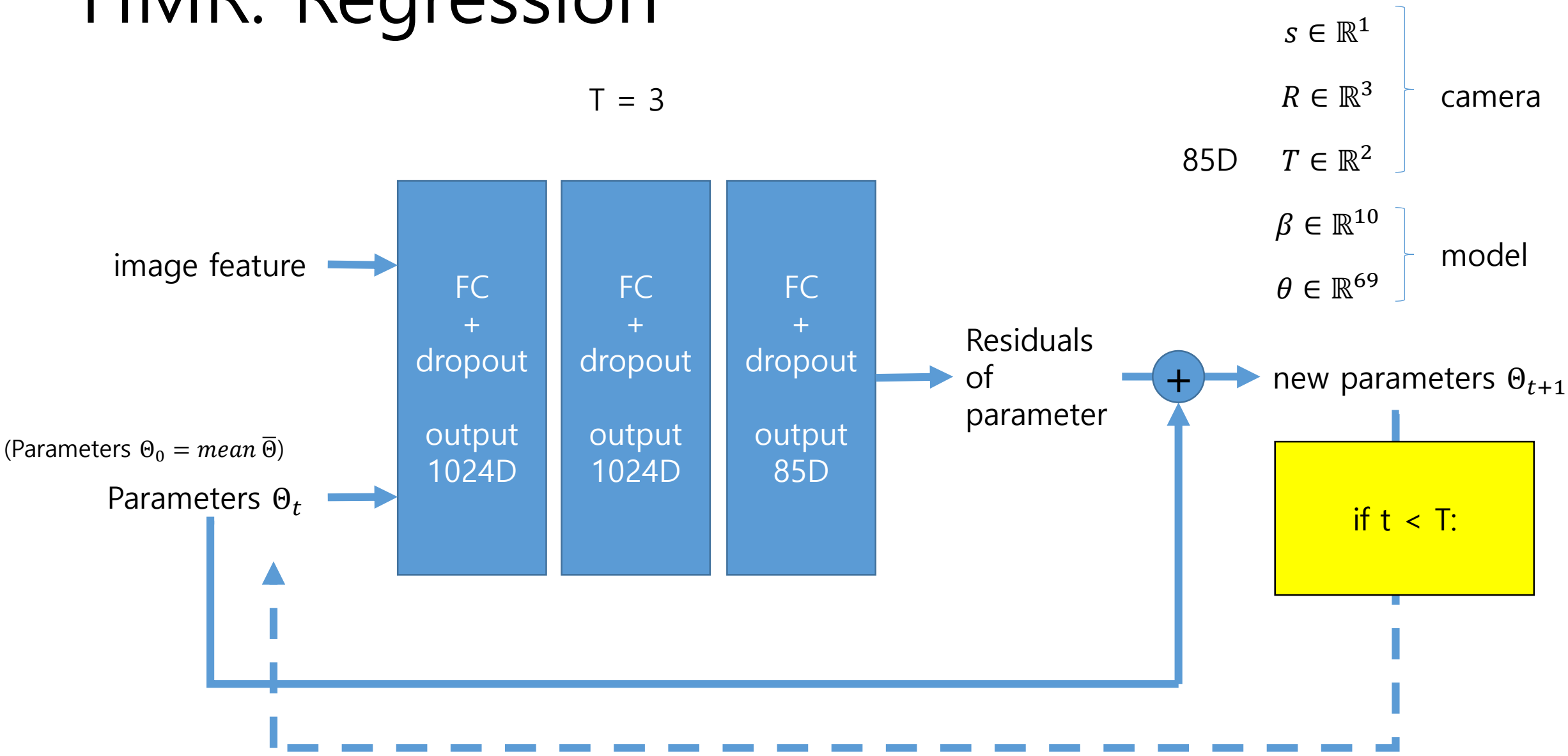
- If this is not the case (e.g., when changing the input/output channels), we can perform a linear projection Ws by the shortcut connections to match the dimensions.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

HMR: Encoder: Resnet-50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

HMR: Regression



HMR: Loss

$$L = \lambda(L_{\text{reproj}} + \mathbb{1}L_{\text{3D}}) + L_{\text{adv}}$$

- λ controls the relative importance of each objective

$L_{\text{reproj}} = \Sigma_i ||v_i(\mathbf{x}_i - \hat{\mathbf{x}}_i)||_1$ 2d joints

- 1 is an indicator function that is 1 if ground truth 3D is available for an image and 0 otherwise.

$$L_{\text{3D}} = L_{\text{3D joints}} + L_{\text{3D smpl}}$$

$L_{\text{joints}} = ||(\mathbf{X_i} - \hat{\mathbf{X}}_\mathbf{i})||_2^2$ 3d joints

$L_{\text{smpl}} = ||[\boldsymbol{\beta}_i, \boldsymbol{\theta}_i] - [\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\theta}}_i]||_2^2$ model parameter

$$L_{\text{adv}}(E) = \sum_i \mathbb{E}_{\Theta \sim p_E}[(D_i(E(I)) - 1)^2]$$

HMR: Loss

- For each iteration, given Θ_t, **if t == T:** $L = \lambda(L_{\text{reproj}} + \mathbb{1}L_{\text{3D}}) + L_{\text{adv}}$

else: $L = L_{\text{adv}}$

HMR: Loss: Discriminator

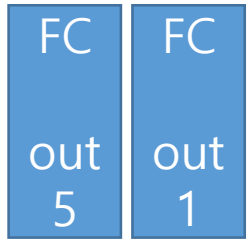
- K + 2 discriminators
  - Discriminator for betas (input: $\beta \in \mathbb{R}^{10}$)
  - K discriminators for each joint (input: $\mathbb{R}^{9}$)
  - Discriminator for all joints (input: $\theta \in \mathbb{R}^{9K}$)

- SMPL has a factorized form that we can take advantage of to make the adversary more data efficient and stable to train.

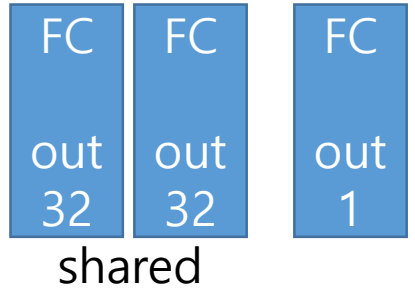
- Especially, the pose is based on a kinematic tree, so we further decompose the pose discriminators and train one for each joint rotation.

HMR: Loss: Discriminator

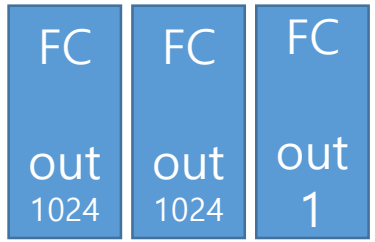
- Discriminator for betas (input: $\beta \in \mathbb{R}^{10}$)



- K discriminators for each joint (input: $\mathbb{R}^{9}$)



- Discriminator for all joints (input: $\theta \in \mathbb{R}^{9K}$)



HMR: Loss: Discriminator

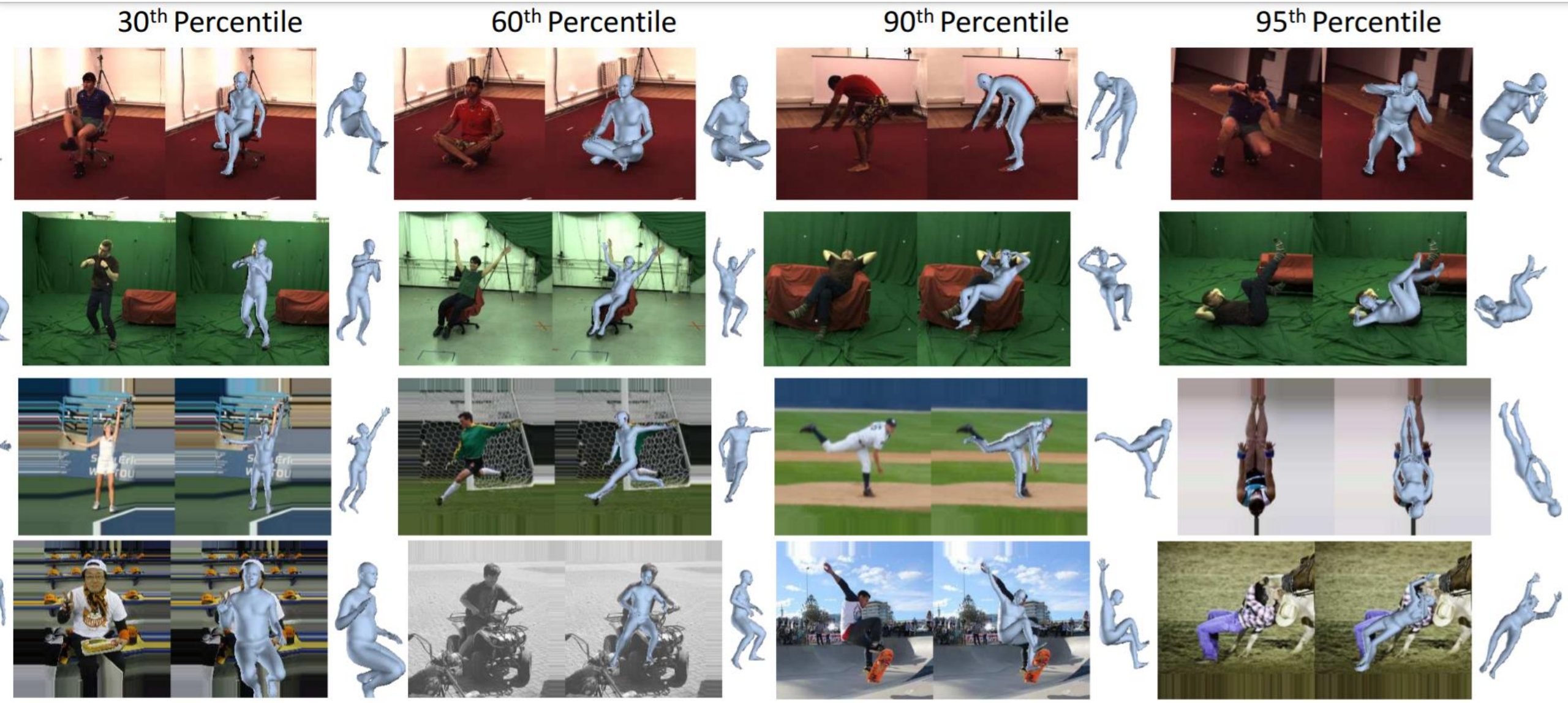
- Loss function for encoder E (including the image encoder and the 3D module)

$$L_{\text{adv}}(E) = \sum_i \mathbb{E}_{\Theta \sim p_E}[(D_i(E(I)) - 1)^2]$$

$D_i(E(I))$

- Loss function for each discriminator D

$$L(D_i) = \mathbb{E}_{\Theta \sim p_{\text{data}}}[(D_i(\Theta) - 1)^2] + \mathbb{E}_{\Theta \sim p_E}[D_i(E(I))^2]$$


30th Percentile
60th Percentile
90th Percentile
95th Percentile
Regression error?
Encoder error?

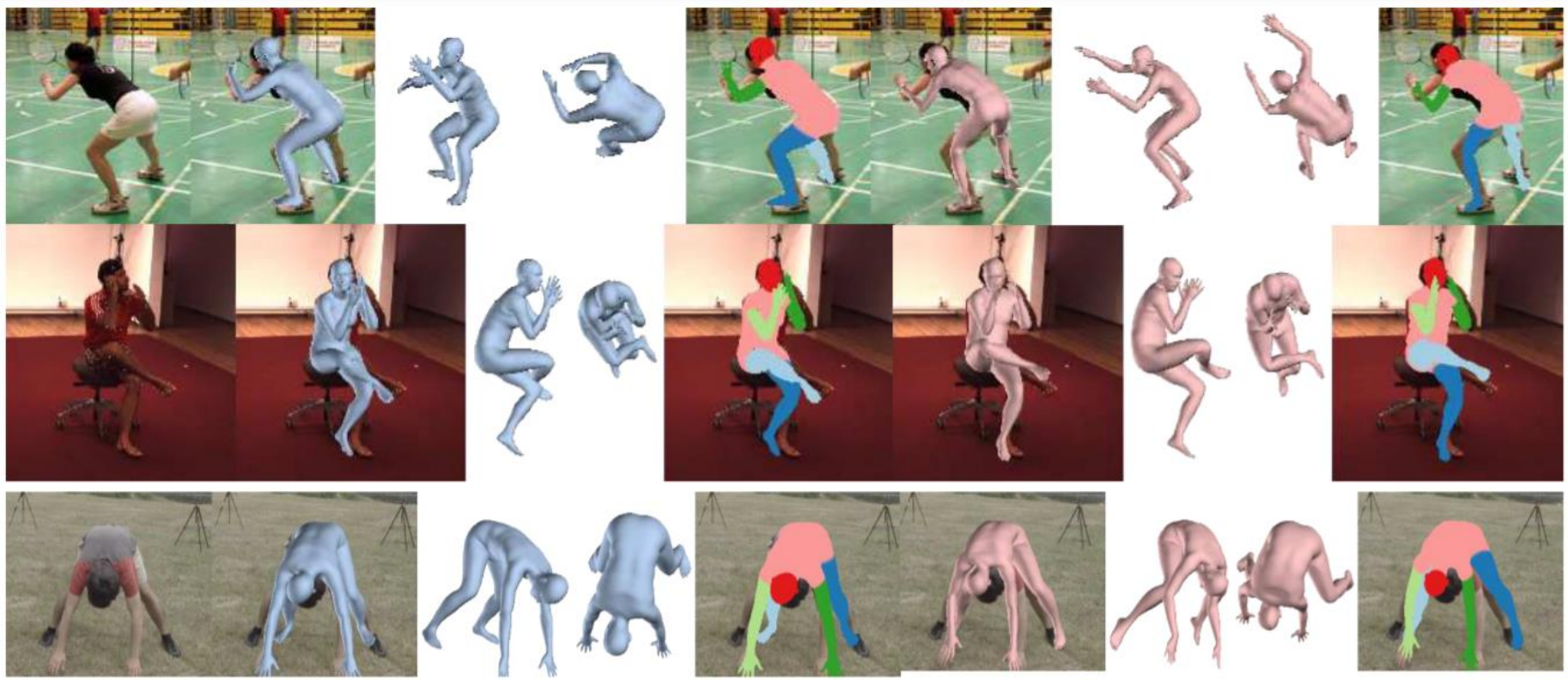


Figure 4: **Results with and without paired 3D supervision.** 3D reconstructions, without direct 3D supervision, are very close to those of the supervised model.

Figure 5: **No Discriminator No 3D.** With neither the discriminator, nor the direct 3D supervision, the network produces monsters. On the right of each example we visualize the ground truth keypoint annotation in unfilled circles, and the projection in filled circles. Note that despite the unnatural pose and shape, its 2D projection error is very accurate.