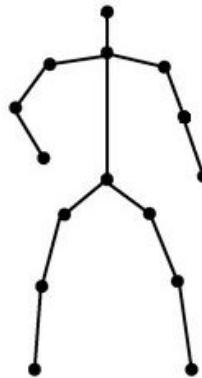


Deep learning advances on human pose and shape estimation

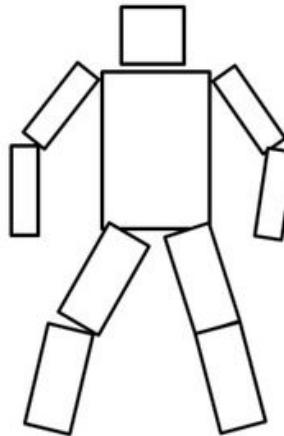
Meysam Madadi

What is articulated human pose?

- Given body kinematic tree, human pose is defined as the vector of joints locations either in 2D (image coordinate in pixels) or 3D (world coordinate in meters).
- Human body pose estimation is a way of representing humans in the images,
- It can be represented and estimated from coarse to fine models.



Kinematic model



Cardboard model

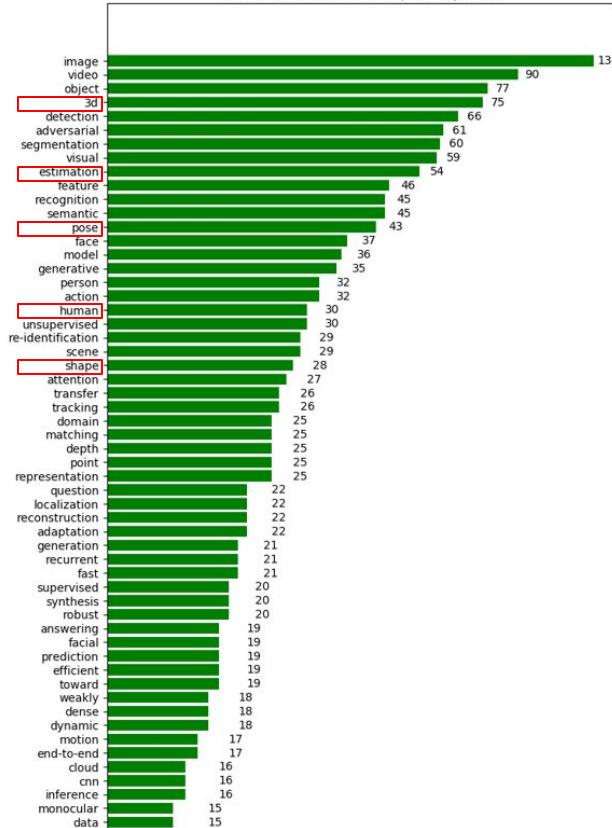


Volumetric/parametric model

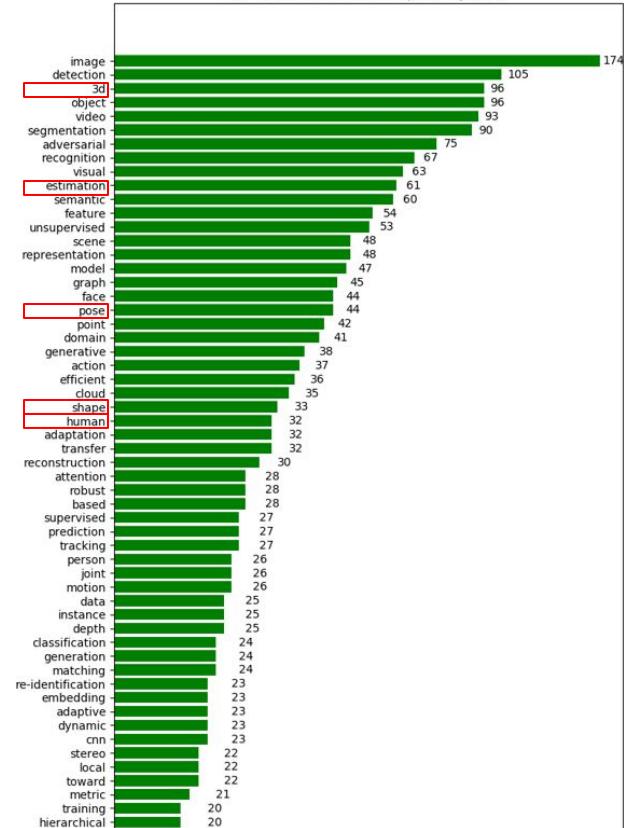
Importance of the pose estimation

Image credit: <https://github.com/hoya012/>

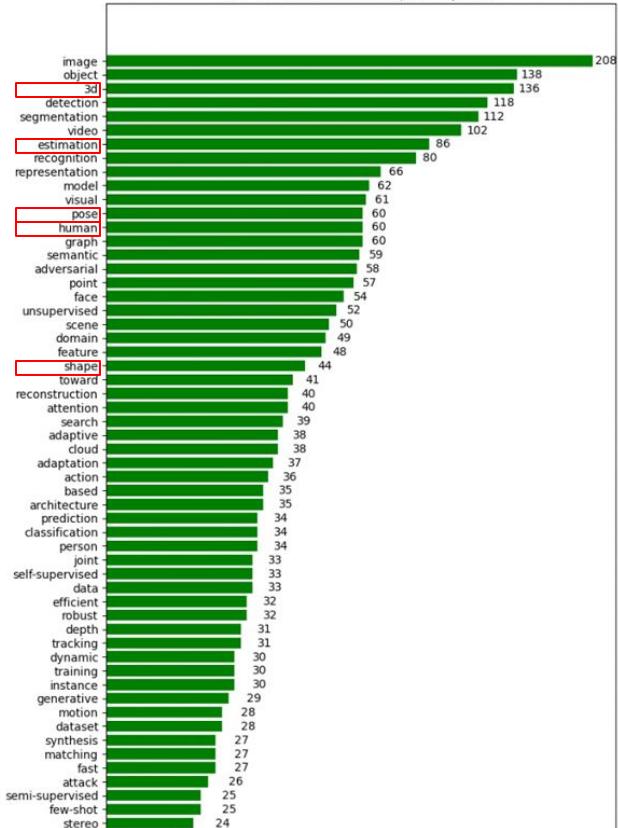
CVPR 2018 Submission Top 75 Keywords



CVPR 2019 Submission Top 75 Keywords



CVPR 2020 Submission Top 75 Keywords



Applications



Movies



Virtual Reality



Pose-based Games



HCI



Animation

*Human Pose
Estimation*



Mental Development



Sports Action Analysis



Surveillance



Military

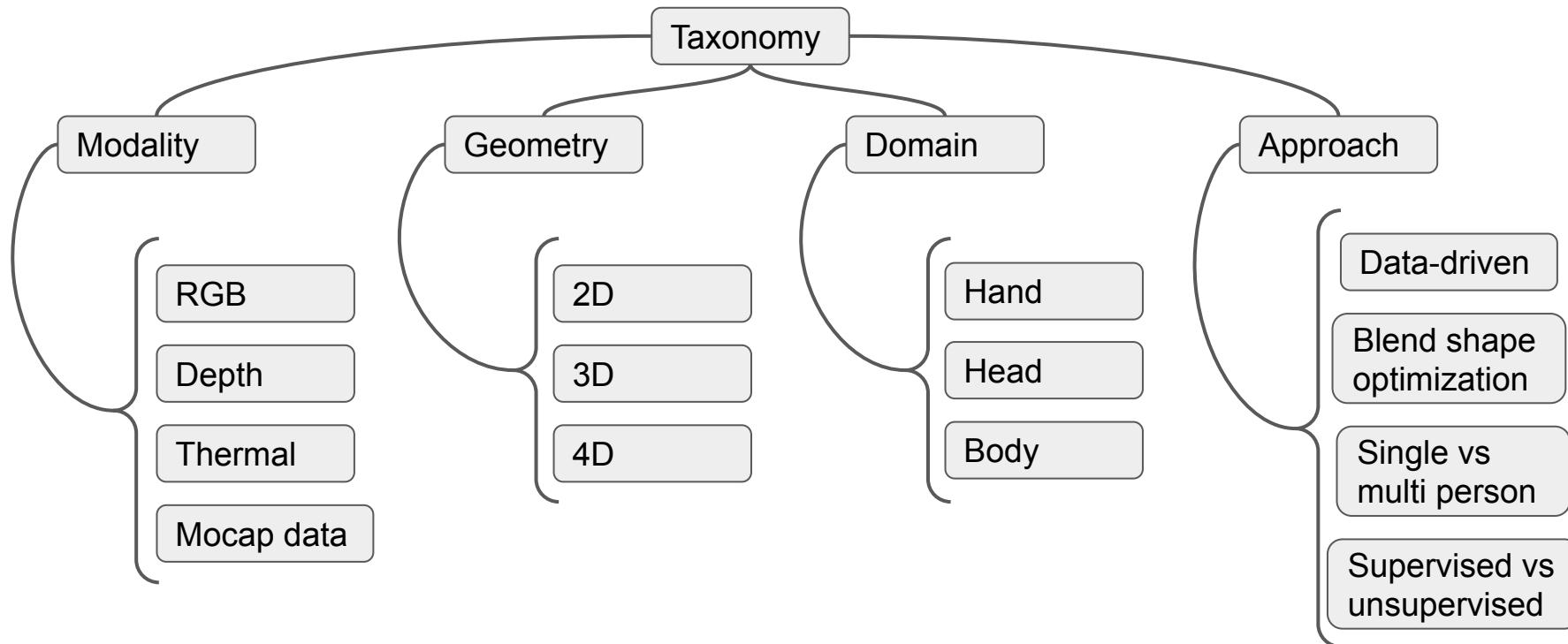


Physiotherapy

Challenges



Taxonomy of pose estimation

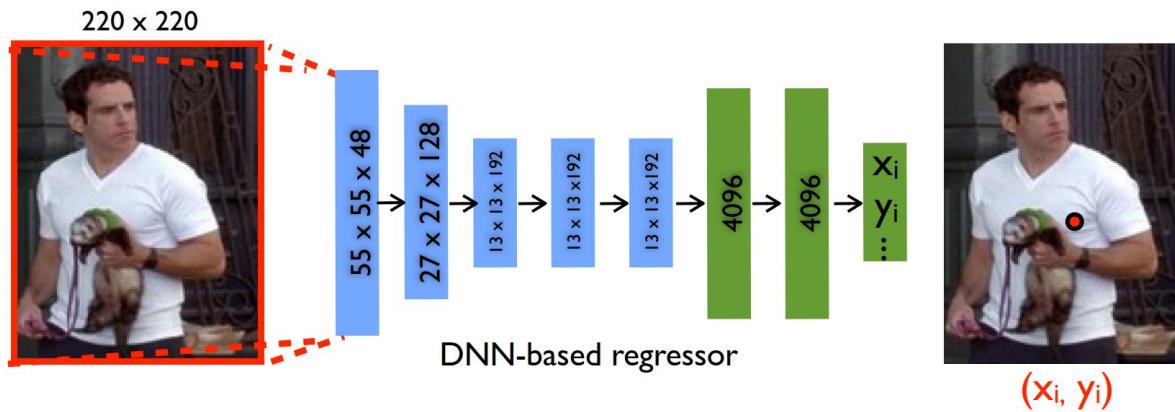


Outlines

- 2D human body pose estimation,
 - Pose regression,
 - Heatmap-based solutions and cascading,
 - Bottom-up and top-down inference.
- 3D human body pose estimation,
 - Lifting 2D to 3D,
 - Volumetric heatmaps and soft-argmax.
- 3D human body pose and shape estimation,
 - Unsupervised estimation,
 - Pose and shape estimation from intermediate representations.
- 3D hand pose estimation,
 - Voxel to voxel prediction.

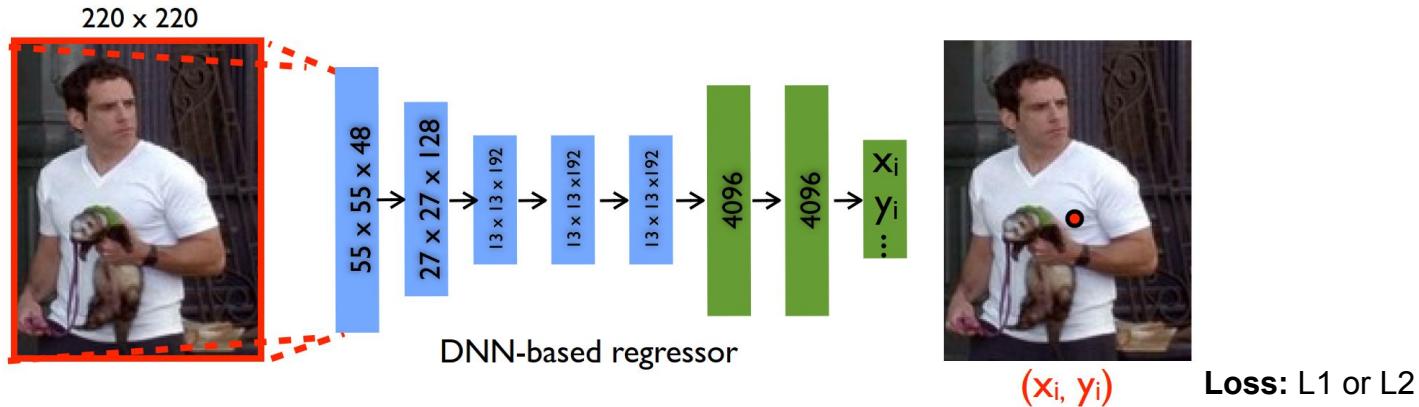
Human pose regression

- The output of the network is directly a vector of (x,y) joints locations,
- This is a difficult task for the network because
 - Pose vector is a highly nonlinear variable,
 - Network must deal with scale and translation as well.



Human pose regression

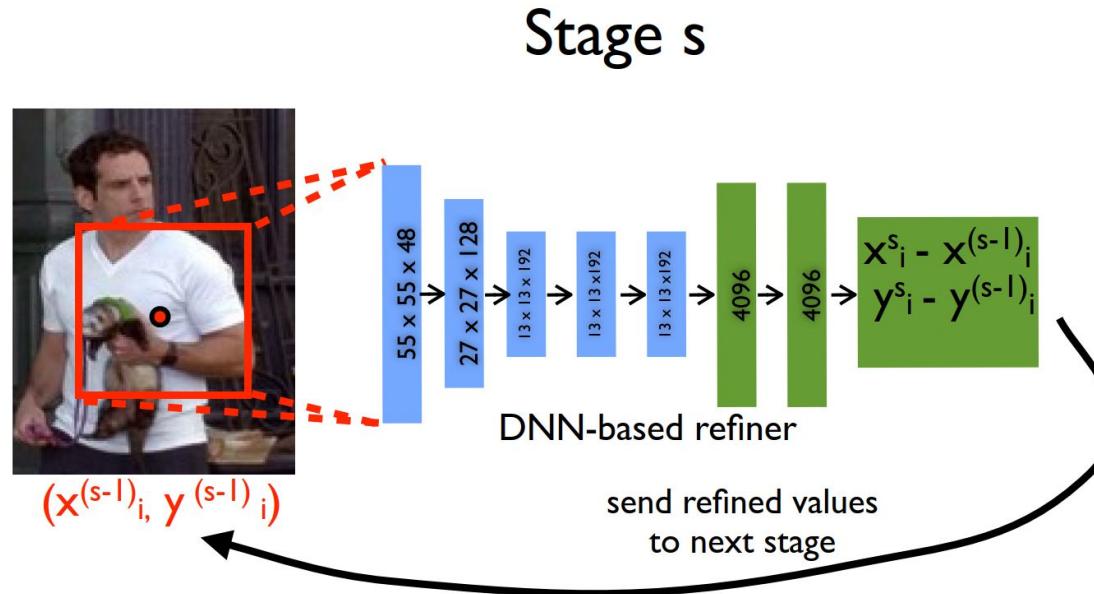
- The output of the network is directly a vector of (x,y) joints locations,
 - This is a difficult task for the network because  **A better alternative:** a dense probability map for each joint.
 - Pose vector is a highly nonlinear variable,  **Solution:** cascade of pose regressors.
 - Network must deal with scale and translation as well.
-  **Solution:** crop person bounding box as pre-processing.



Architecture: AlexNet

Cascade of pose regressors

- Given an initial estimation of the joints (in holistic view), iteratively refines the error (in local view).



Cascade of pose regressors

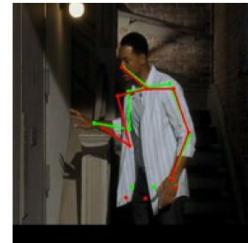
Limitations:

- Sensitive to initial estimation,
- Local solution, easily can stick to local minima,
- Lack of structured predictions.

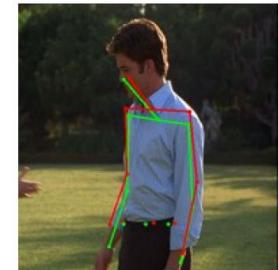
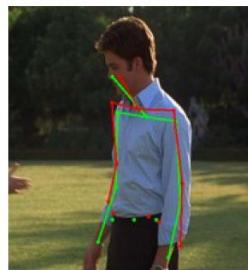
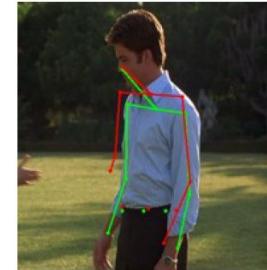
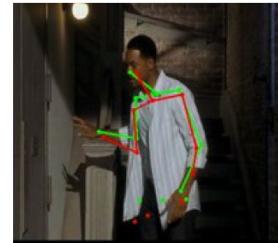
Initial stage 1



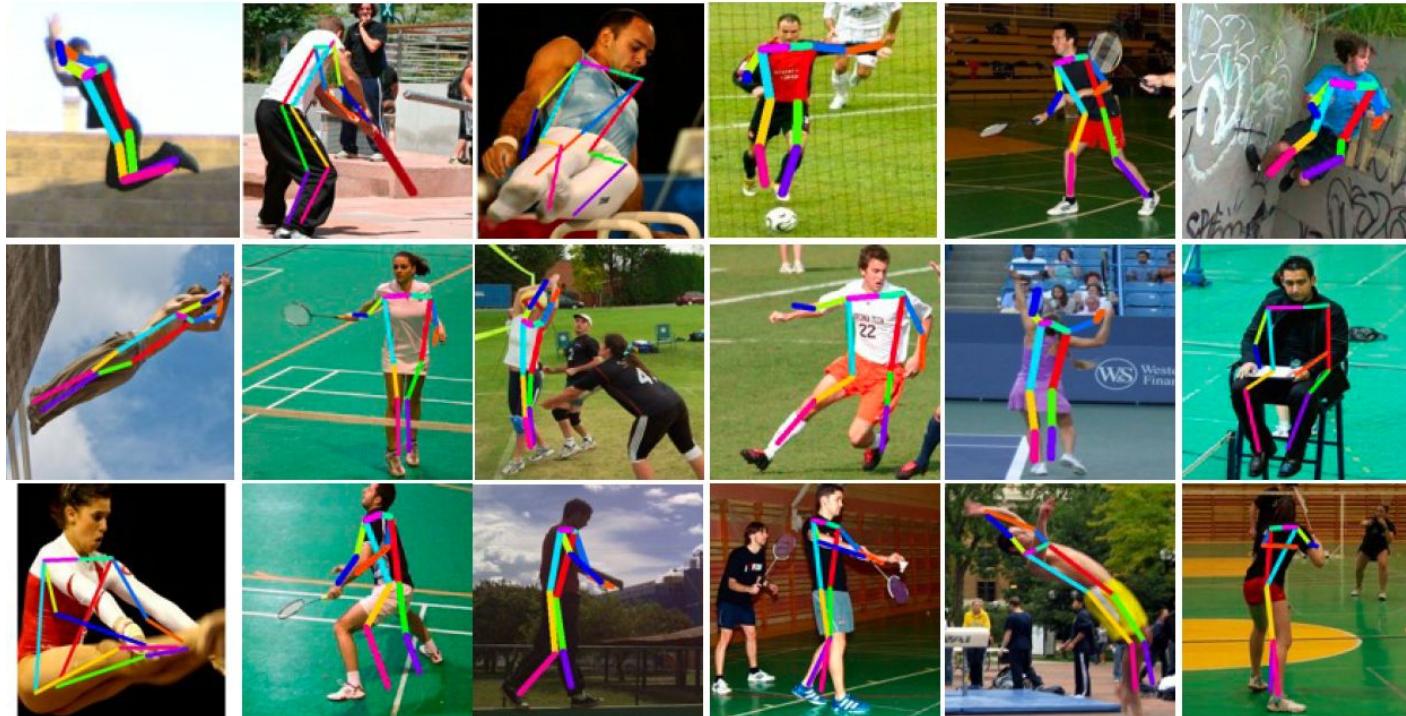
stage 2



stage 3



Results on LSP dataset

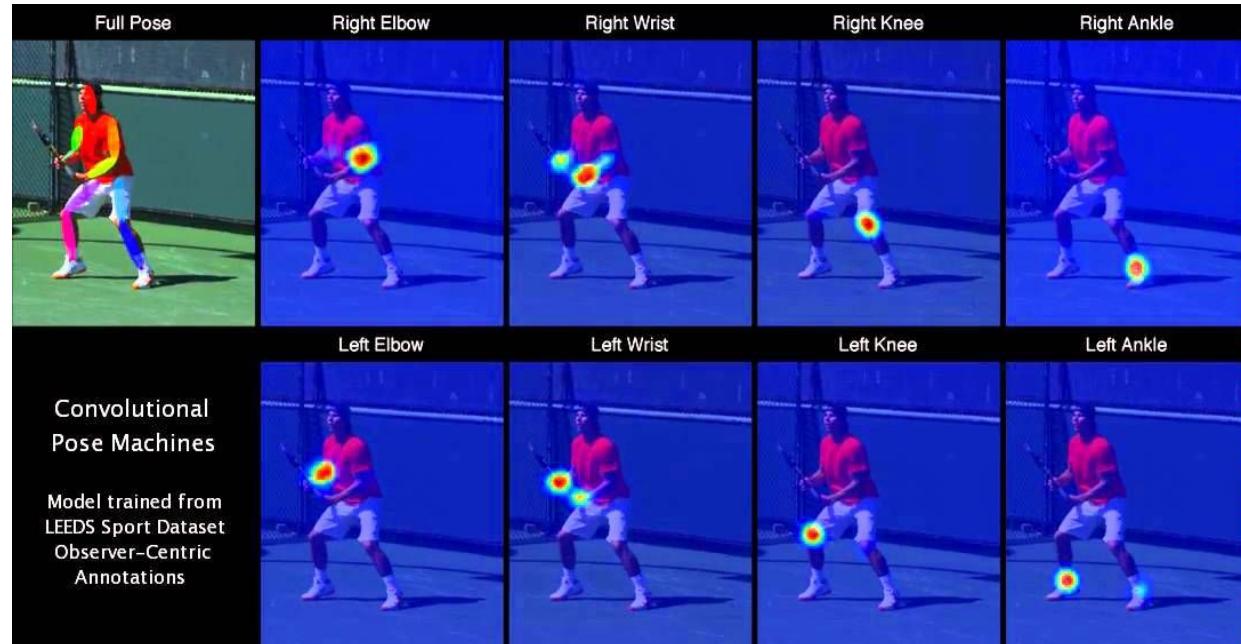


Results on LSP dataset



Joints heatmaps instead of pose vector

- It is an easier problem for the network,
- Still can be combined with pose regressors,
- The map can be fed into graphical models to learn higher order joint relationships.



Context is important

Which part belongs to a human body?



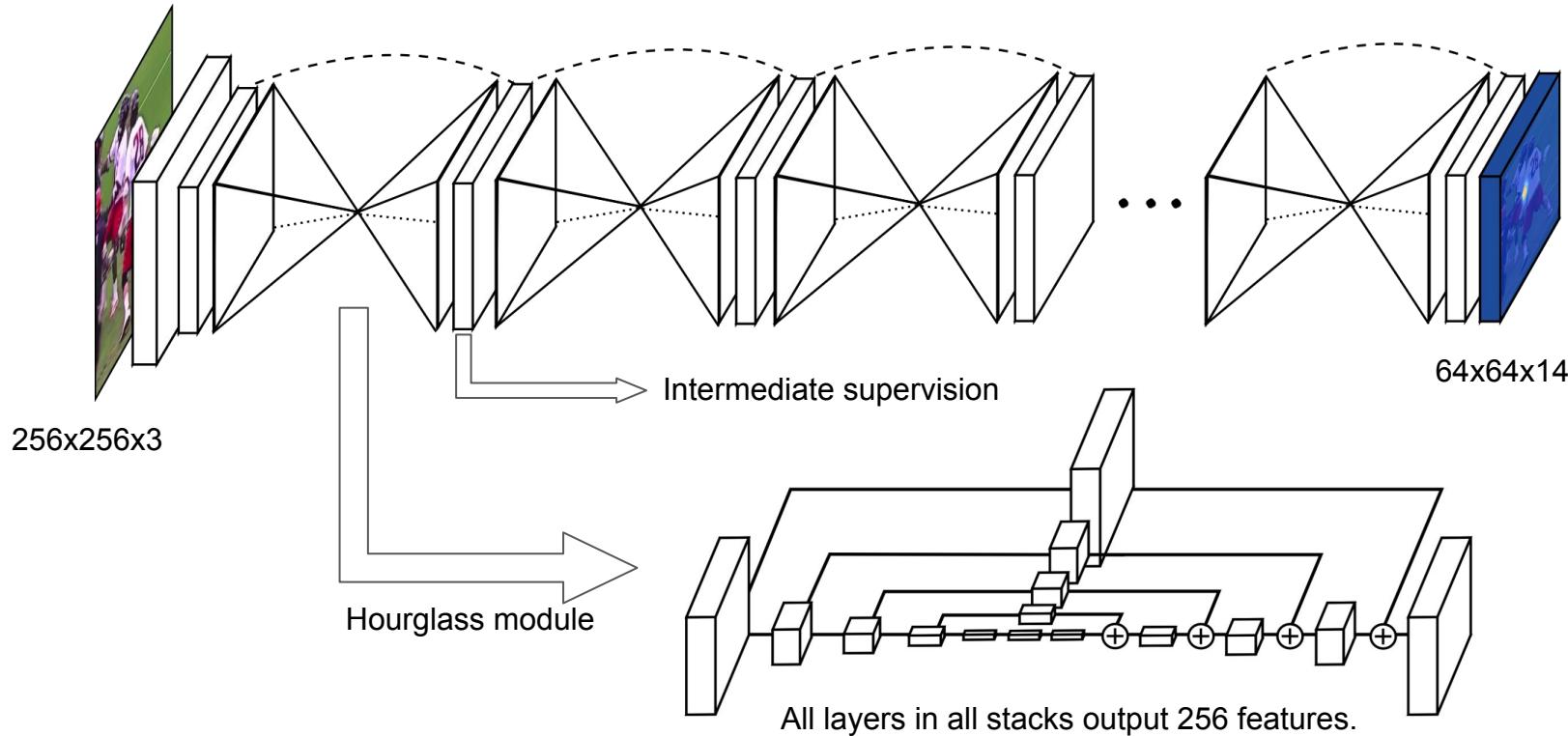
Context is important

Which part belongs to a human body?

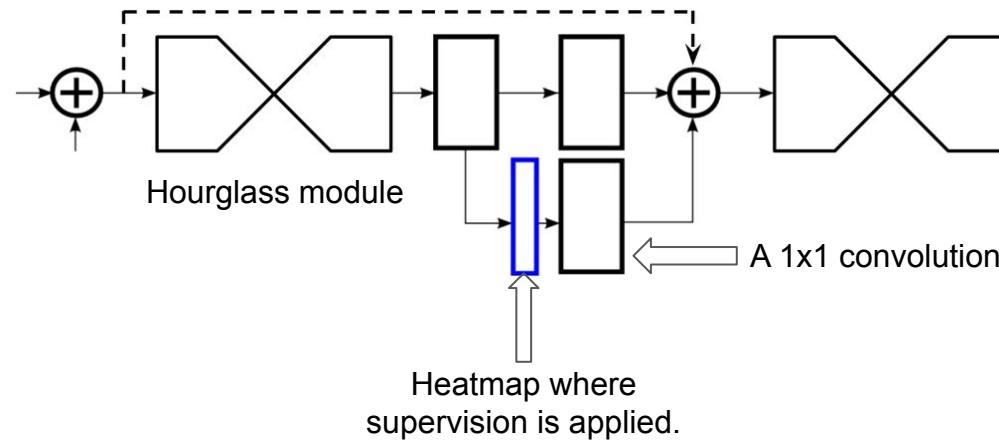
- Local evidence is weak,
- Larger receptive field = more context,
- Recover from failures by cascading.



Stacked Hourglass Network (SHN)



Stacked Hourglass Network: intermediate supervision

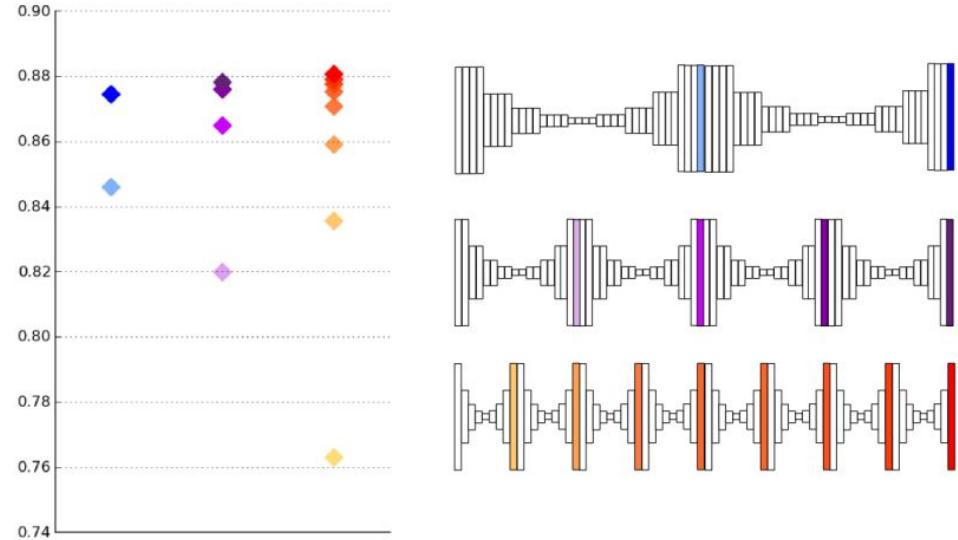


Stacked Hourglass Network

Cons:

- Quite heavy in GPU,
- Joints may be confused with background,
- Still does not explicitly deal with structure.

Intermediate Prediction Accuracy (Validation, PCKh@0.5)

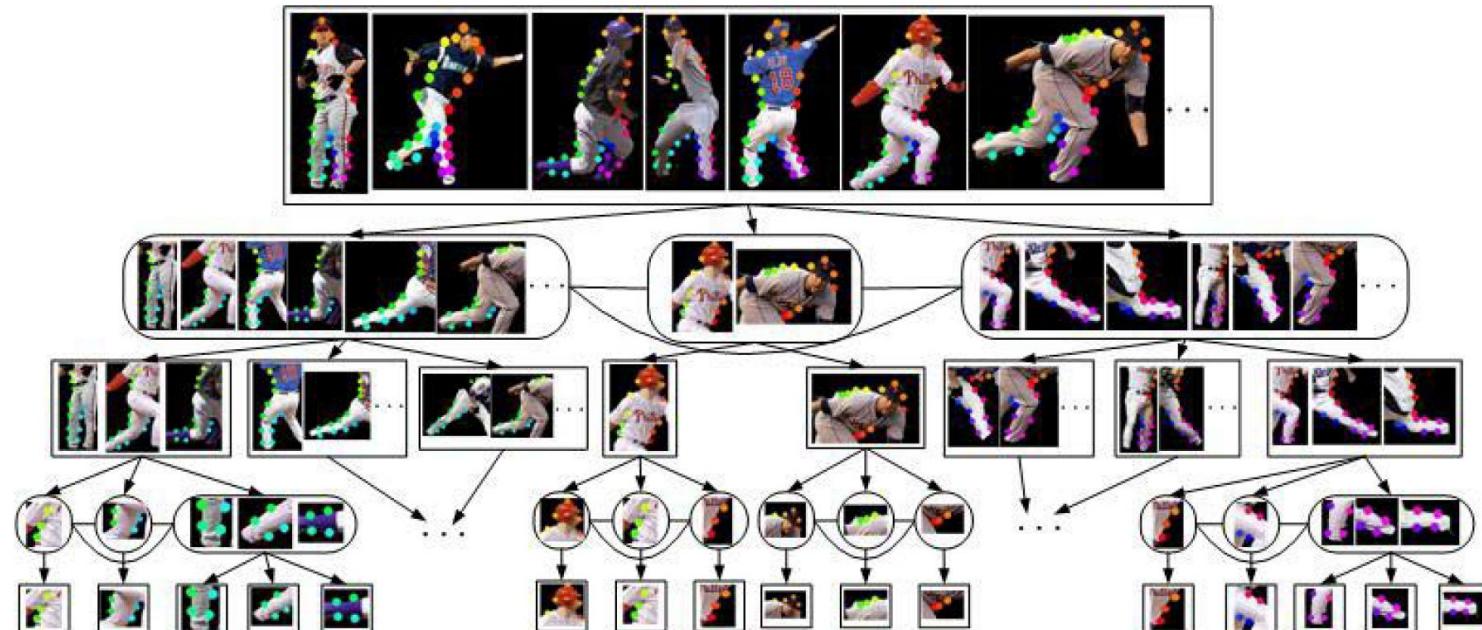


Some results on MPII dataset



AND-OR graph as body part representation

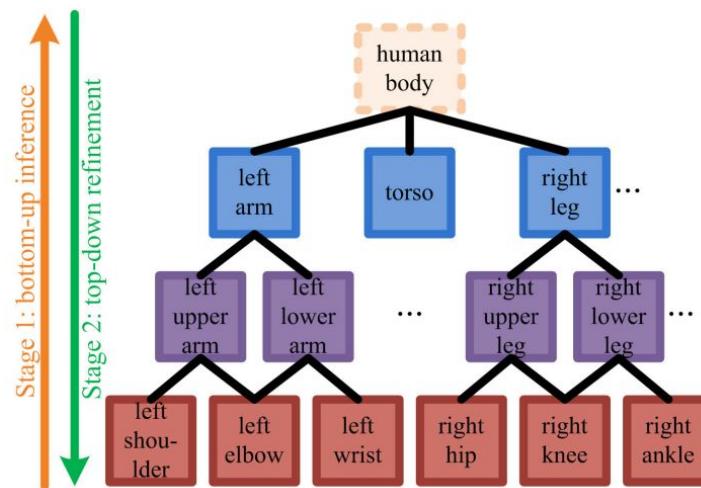
- The goal is to optimize a model to find the best possible combination of leaf nodes,
- Body representation as a hierarchy satisfies some articulation constraints.



Deeply learned compositional models

1. We have a compositional model of part-subpart relationships,
2. Traditional solutions by modelling the graph (or tree) with Gibbs formulation,
3. We want a bottom-up and top-down inference,

Question: How to model it with CNN models?



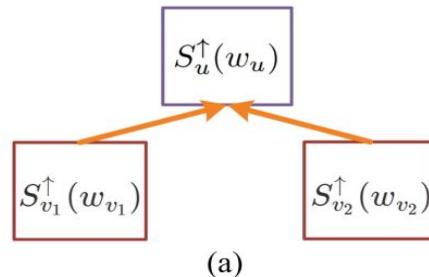
$$p(\Omega|\mathbf{I}) = \frac{1}{Z} \exp\{-E(\Omega, \mathbf{I})\}$$

$$S(\Omega) \equiv -E(\Omega, \mathbf{I}) = \sum_{u \in \mathcal{V}^{leaf}} \phi_u^{leaf}(w_u, \mathbf{I}) + \sum_{u \in \mathcal{V}^{and}} \sum_{v \in ch(u)} \phi_{u,v}^{and}(w_u, w_v)$$

Score to be minimized
 ↑
 Unary term
 ↑
 Pairwise term

First define updating rule

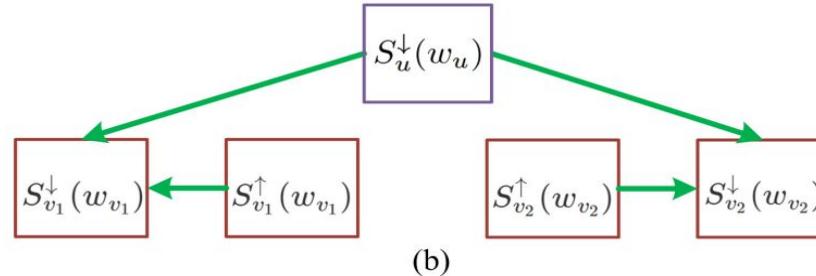
Bottom-up inference:



$$(\text{Leaf}) \quad S_u^{\uparrow}(w_u) = \phi_u^{\text{leaf}}(w_u, \mathbf{I})$$

$$(\text{And}) \quad S_u^{\uparrow}(w_u) = \sum_{v \in \text{ch}(u)} \max_{w_v} [\phi_{u,v}^{\text{and}}(w_u, w_v) + S_v^{\uparrow}(w_v)]$$

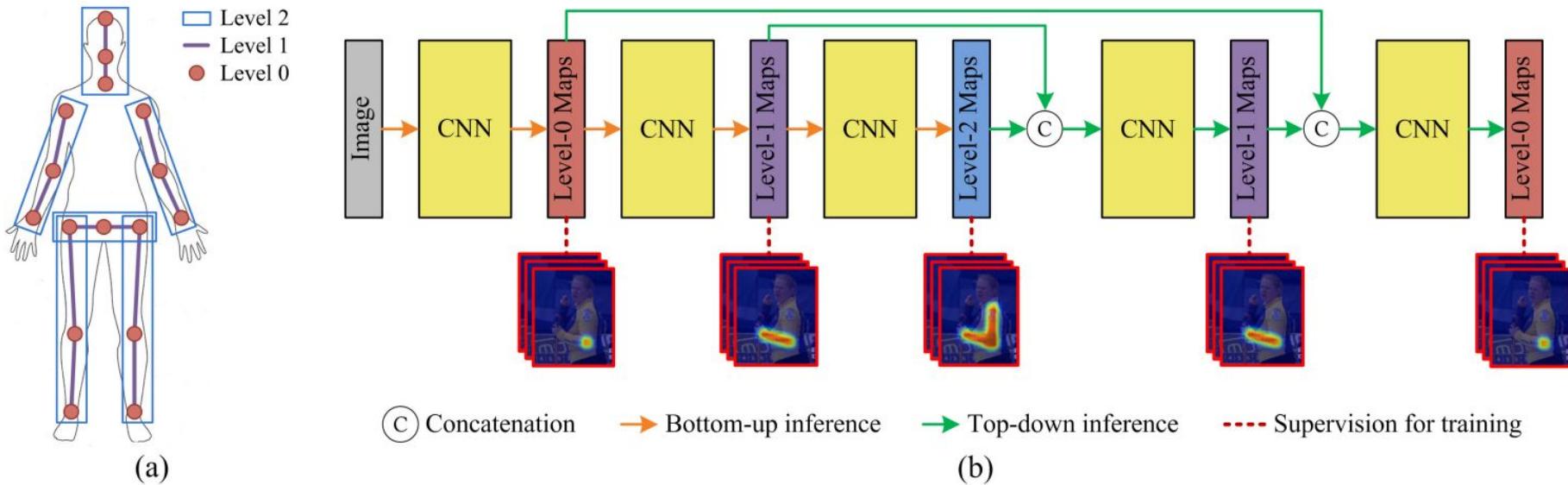
Top-down inference:



$$(\text{Root}) \quad w_u^* = \operatorname{argmax}_{w_u} S_u^{\downarrow}(w_u) \equiv \operatorname{argmax}_{w_u} S_u^{\uparrow}(w_u)$$

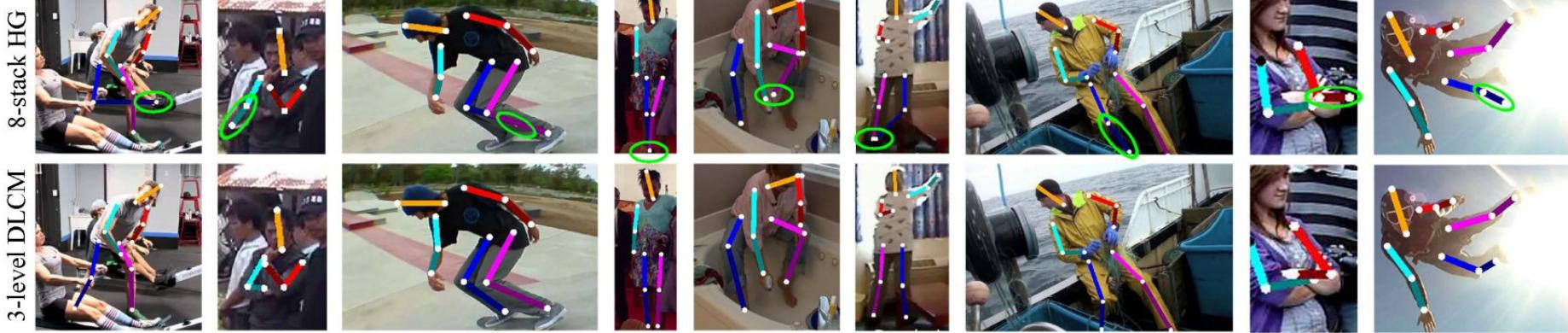
$$(\text{Non-root}) \quad w_v^* = \operatorname{argmax}_{w_v} S_v^{\downarrow}(w_v) \equiv \operatorname{argmax}_{w_v} [\phi_{u,v}^{\text{and}}(w_u^*, w_v) + S_v^{\uparrow}(w_v)]$$

Revisit stacked hourglass network



Heatmaps of level i ($i > 0$) are composed of heatmaps of level 0 and level i .

Some results comparing to the 8-stack hourglass



Further reading

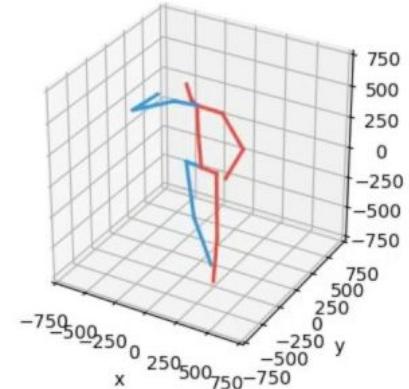
- Multi-context attention maps:
http://openaccess.thecvf.com/content_cvpr_2017/papers/Chu_Multi-Context_Attention_for_CVPR_2017_paper.pdf
- End-to-end training of graphical models and CNN: www.ee.cuhk.edu.hk/~xgwang/papers/yangOLWcvpr16.pdf
- Pose estimation by the help of generative adversarial training:
<http://lamda.nju.edu.cn/weixs/publication/iccv17.pdf>

3D human body pose

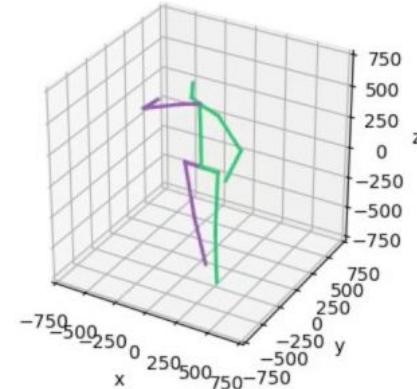
- 3D pose is the vector of body joints in 3D space,
- 2D pose is the projection of 3D pose to image plane,
- 3D pose is defined with the same models as 2D pose,
- 3D pose estimation is a challenging task since depth dimension is lost during projection to RGB image.



2d observation



3d ground truth



3d prediction

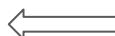
3D human body pose estimation

Solutions:

- Directly regress 3D pose from RGB image,
- Estimate 2D pose and then regress 3D pose from 2D,
- Apply volumetric heatmaps and estimate 3D pose,
- Apply multi-task learning by the use of different modalities.

3D human body pose estimation

Solutions:

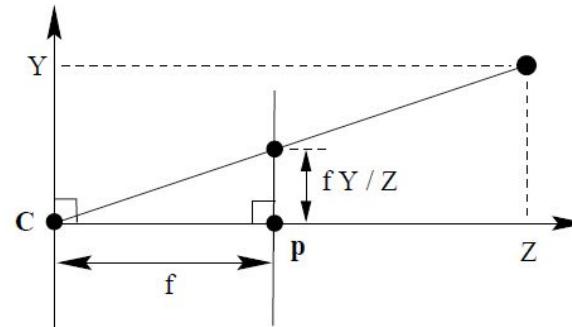
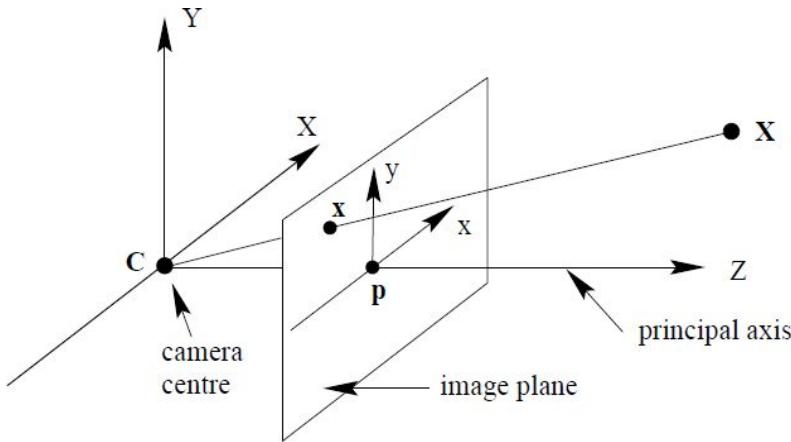
- Directly regress 3D pose from RGB image,  A difficult problem to the network similar to 2D pose.
- Estimate 2D pose and then regress 3D pose from 2D,
- Apply volumetric heatmaps and estimate 3D pose,
- Apply multi-task learning by the use of different modalities.  Annotating data for all modalities is not a trivial task for many datasets.

3D human body pose estimation

Solutions:

- Directly regress 3D pose from RGB image,
- Estimate 2D pose and then regress 3D pose from 2D,
- Apply volumetric heatmaps and estimate 3D pose,
- Apply multi-task learning by the use of different modalities.

Perspective projection



f : focal length
 p : image center

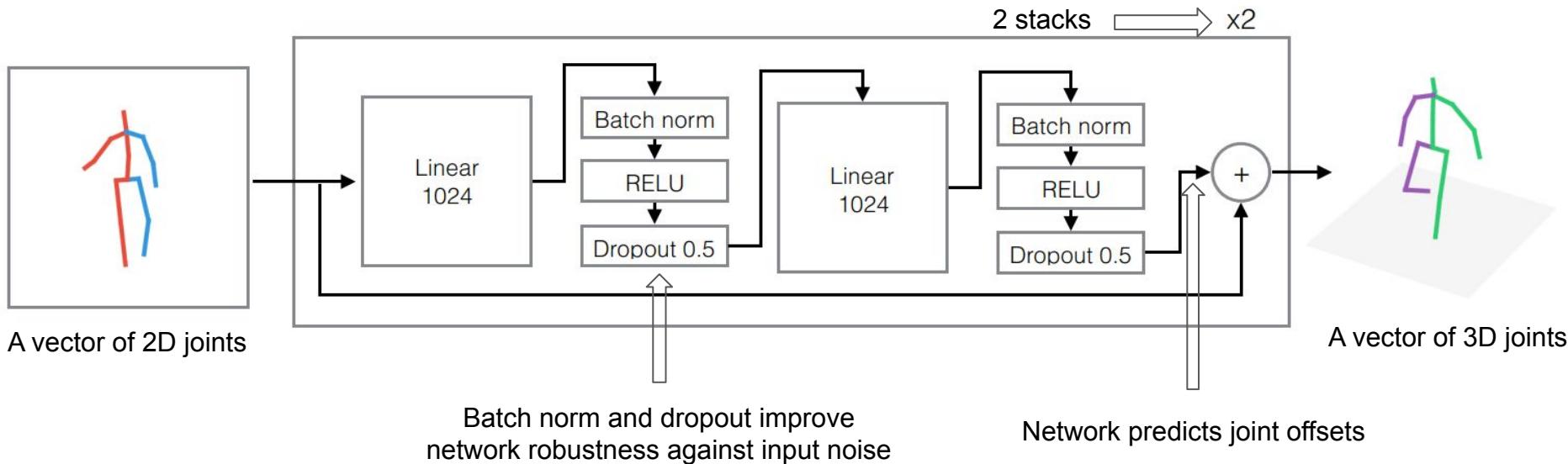
$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Projection matrix

$$\begin{aligned} \text{3D to 2D} &>>> [u \ v \ 1]^T = K [X/Z \ Y/Z \ 1]^T \\ \text{2D to 3D} &>>> [X \ Y \ Z]^T = K^{-1} [uZ \ vZ \ Z]^T \end{aligned}$$

$$\begin{aligned} u &= f_x X/Z + p_x \\ v &= f_y Y/Z + p_y \end{aligned}$$

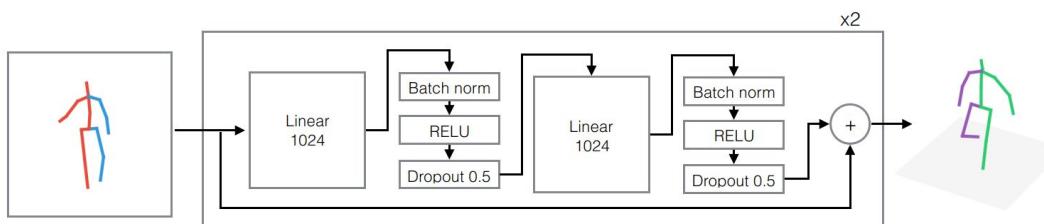
Lifting 2D joints to 3D: A simple solution



Lifting 2D joints to 3D: A simple solution

Limitations:

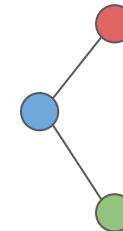
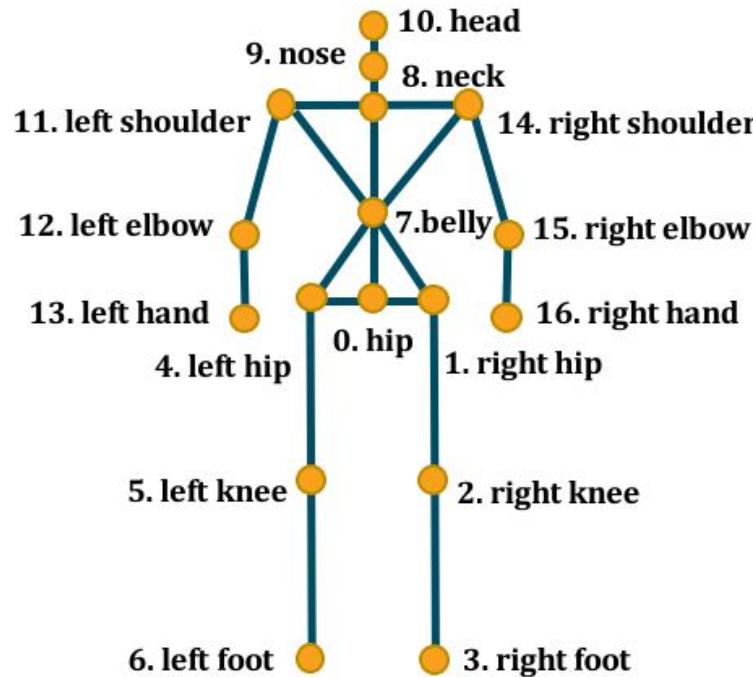
- This is an ill-posed problem, because many 3D poses can be projected to the same 2D pose,
- If 2D pose and 3D pose are not trained jointly, then solution is suboptimal. Because 2D errors are propagated to 3D.



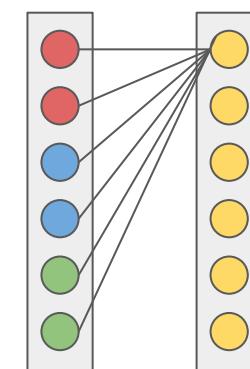
	error (mm)	Δ
Ours	67.5	–
w/o batch norm	88.5	21.0
w/o dropout	71.4	3.9
w/o batch norm w/o dropout	76.0	8.5
w/o residual connections	75.8	8.3
w/o camera coordinates	101.1	33.6
<hr/>		
1 block	74.2	6.7
2 blocks (Ours)	67.5	–
4 blocks	69.3	1.8
8 blocks	69.7	2.4

Table 5. Ablative and hyperparameter sensitivity analysis.

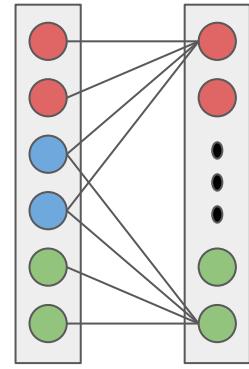
Lifting 2D joints to 3D: Grouping correlated joints



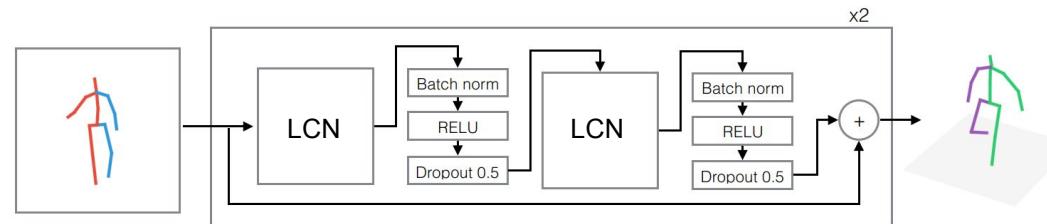
FCN



LCN (Locally Connected Network)



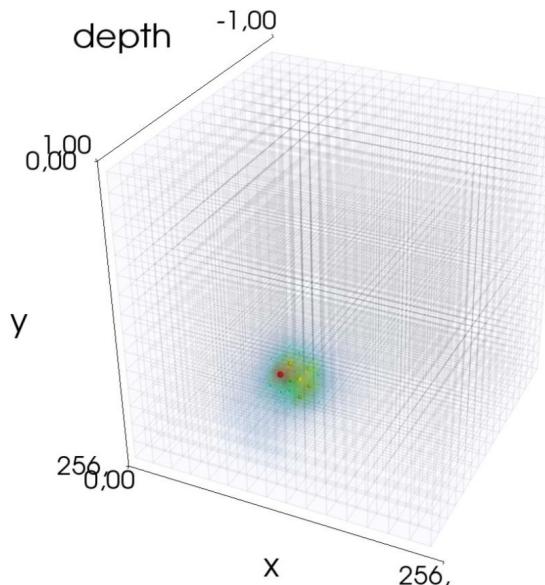
Lifting 2D joints to 3D: Grouping correlated joints



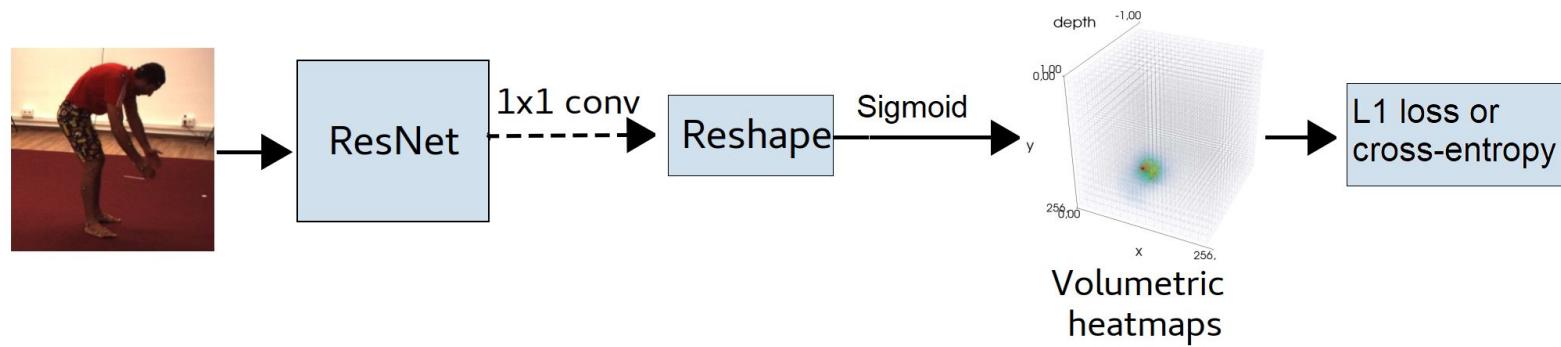
Model	Coordinate System	Error
Martinez <i>et al.</i> [18]	camera CS	67.50
Martinez <i>et al.</i> [18]	pixel CS	63.21
LCN (3-NN)	camera CS	62.95
LCN (3-NN)	pixel CS	57.56

Volumetric heatmaps for 3D pose estimation

- A volumetric heatmap is defined as a tensor in the form of ($\#Width \times \#Height \times \#Depth \times \#Joints$),
- A joint's depth is a continuous value which is discretized into several bins, i.e. $\#Depth$,
- Ground truth heatmap can be defined by a Gaussian.

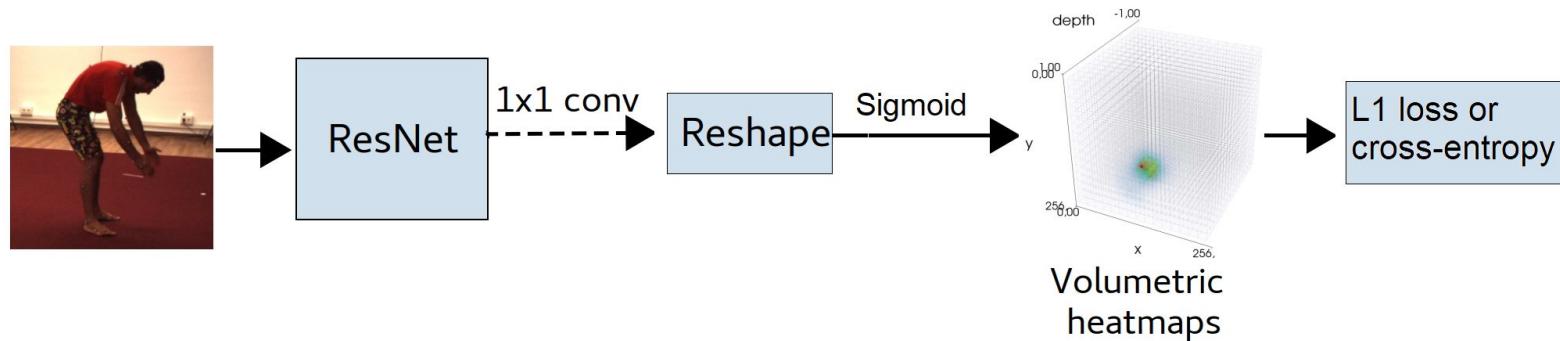


A simple solution



Downside: Depth discretization causes an offset error in joint's depth when *argmax* is applied.

A simple solution



Downside: Depth discretization causes an offset error in joint's depth when *argmax* is applied.

Solution: Using *Soft-argmax* instead of *argmax*.

$$\text{Softmax}(v_{i,j,k}) = \frac{e^{v_{i,j,k}}}{\sum e^v}, v \in \mathbb{R}^{H \times W \times D}$$

$$\text{Soft-argmax}(v) = \sum_i \sum_j \sum_k W_{i,j,k} \text{Softmax}(v_{i,j,k})$$

$$W_{i,j,k} = \left[\frac{i}{H}, \frac{j}{W}, \frac{k}{D} \right]^T$$

A simple solution - data augmentation



2638 occluder objects from Pascal VOC

Filter out 'person', 'truncated', 'difficult' and small object segments



Augmented inputs with pasted occluders

Applied with 50% probability, 1–8 objects, at random scale, at random position

A simple solution - results on H3.6M dataset

Method	Extra pose data in training?	
	no	yes
Sun (ICCV'17) [3]	92.4	59.1
Martinez (ICCV'17) [4]	–	62.9
Zhou (ICCV'17) [5]	–	55.9
Pavlakos (CVPR'18) [6]	71.9	56.2
Sun (ArXiv) [7]	64.1	49.6
Ours (no occlusion augm.)	65.7	–
Ours (full)	55.4	–

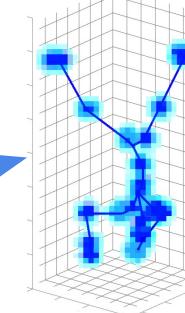
Further reading

- Compositional pose regression: <https://arxiv.org/abs/1704.00159>
- Ordinal depth supervision: <https://arxiv.org/abs/1805.04095>
- Single shot multi person: <https://arxiv.org/abs/1712.03453>
- Temporal estimation: <https://arxiv.org/abs/1811.11742>
- Multi-view: <https://arxiv.org/abs/1803.04775>

Which one do you prefer?



1



2



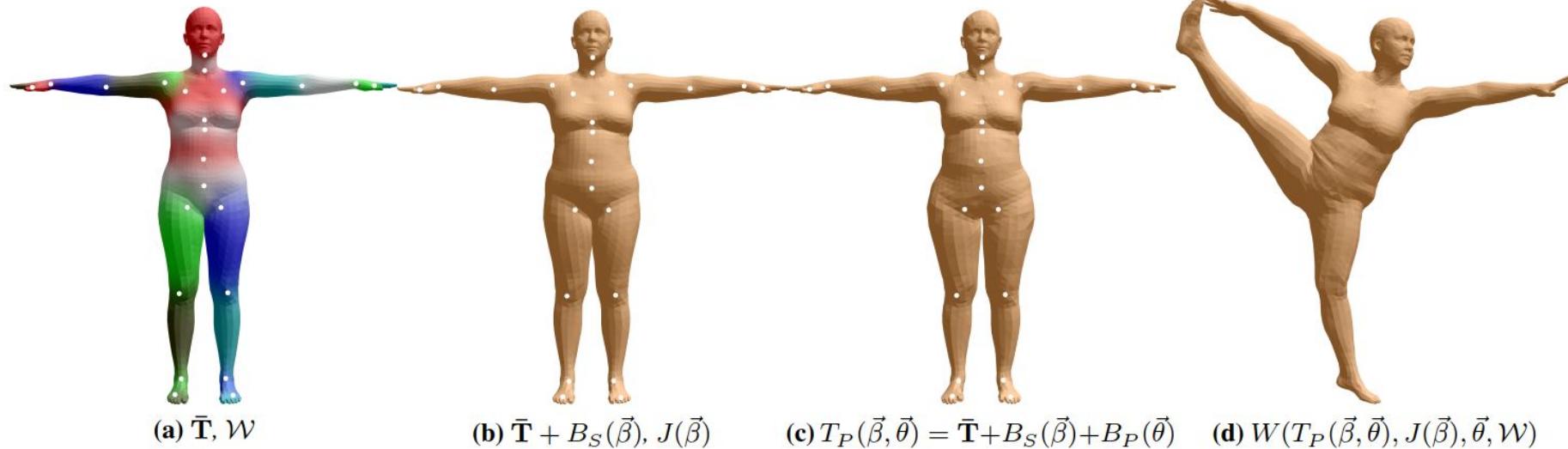
- 3D joints does not represent body anatomy and surface topology,
- We want to estimate a richer representation of body given an RGB image,
- SMPL model has been used for estimation of 3D body joints and mesh.

2D pose estimation

3D pose estimation

3D pose and shape

3D pose and shape by SMPL

Shape β
Pose θ

SMPL

Rendered
3D mesh

2D pose estimation

3D pose estimation

3D pose and shape

A naive solution



CNN

Shape β
Pose θ

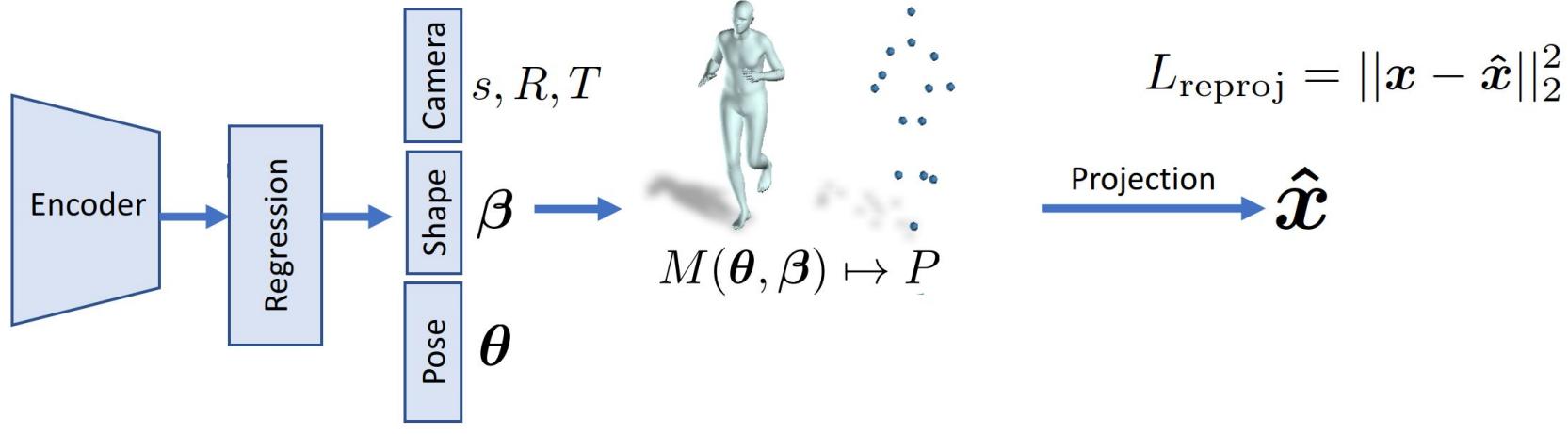
SMPL



Cons

this is a challenging task for the network, plus the noise sensitivity of the model. Besides, direct regression of SMPL parameters may generate artifacts.

Unsupervised pose and shape estimation



Unsupervised pose and shape estimation

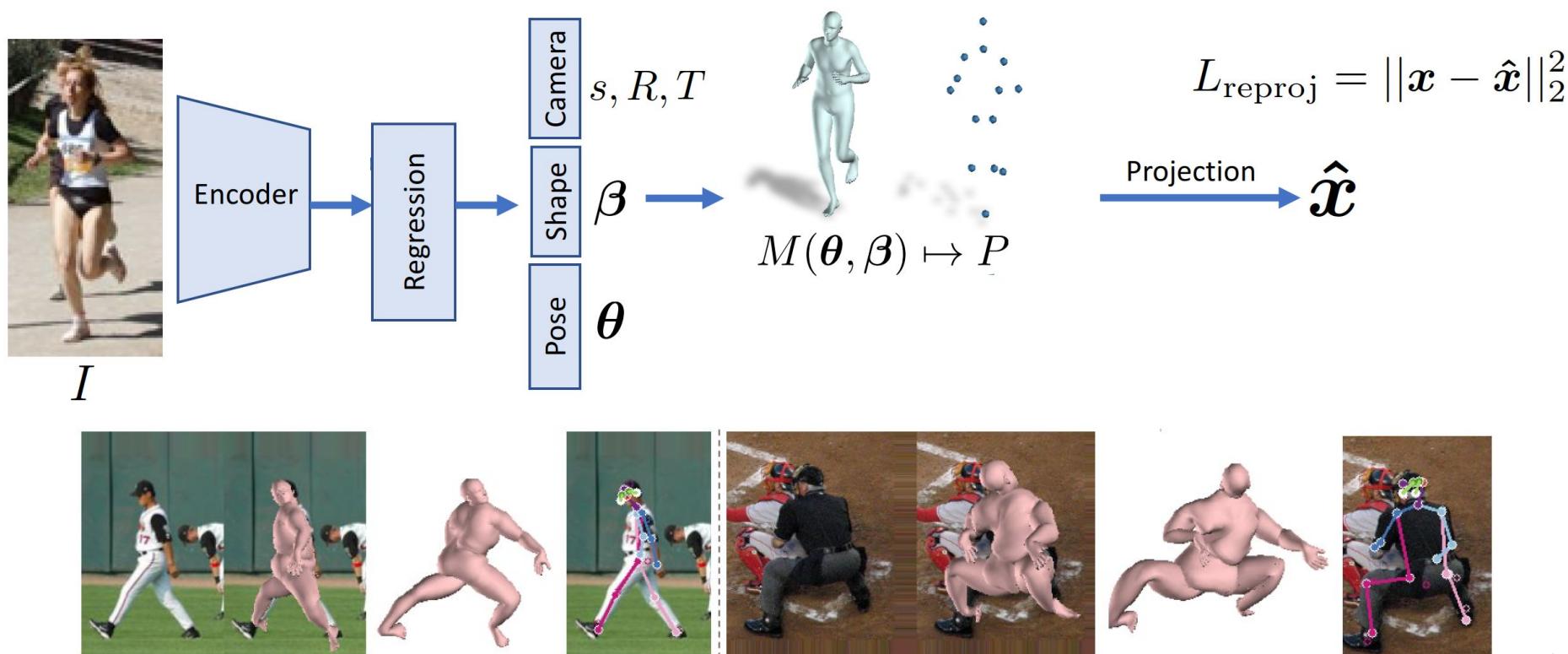


Image credit: <https://arxiv.org/abs/1712.06584>

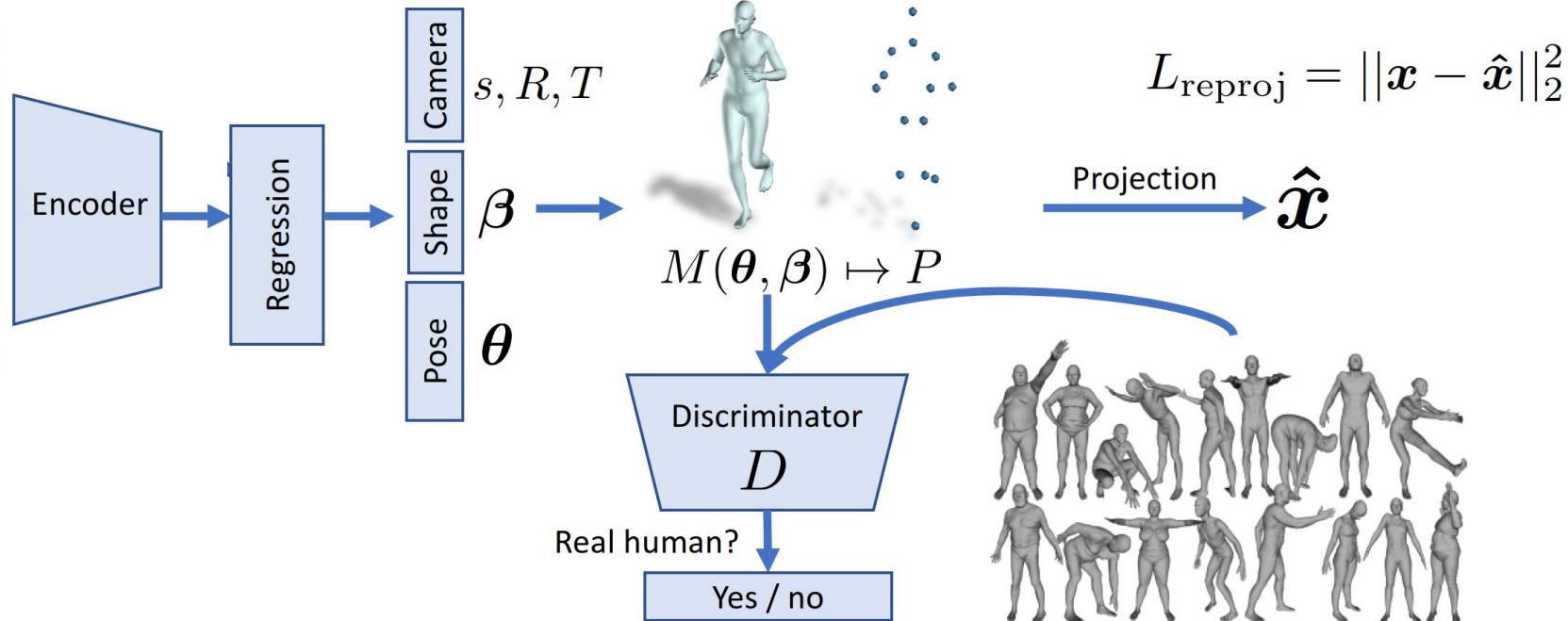
2D pose estimation

3D pose estimation

3D pose and shape

Unsupervised pose and shape estimation

Factorized adversarial prior



2D pose estimation

3D pose estimation

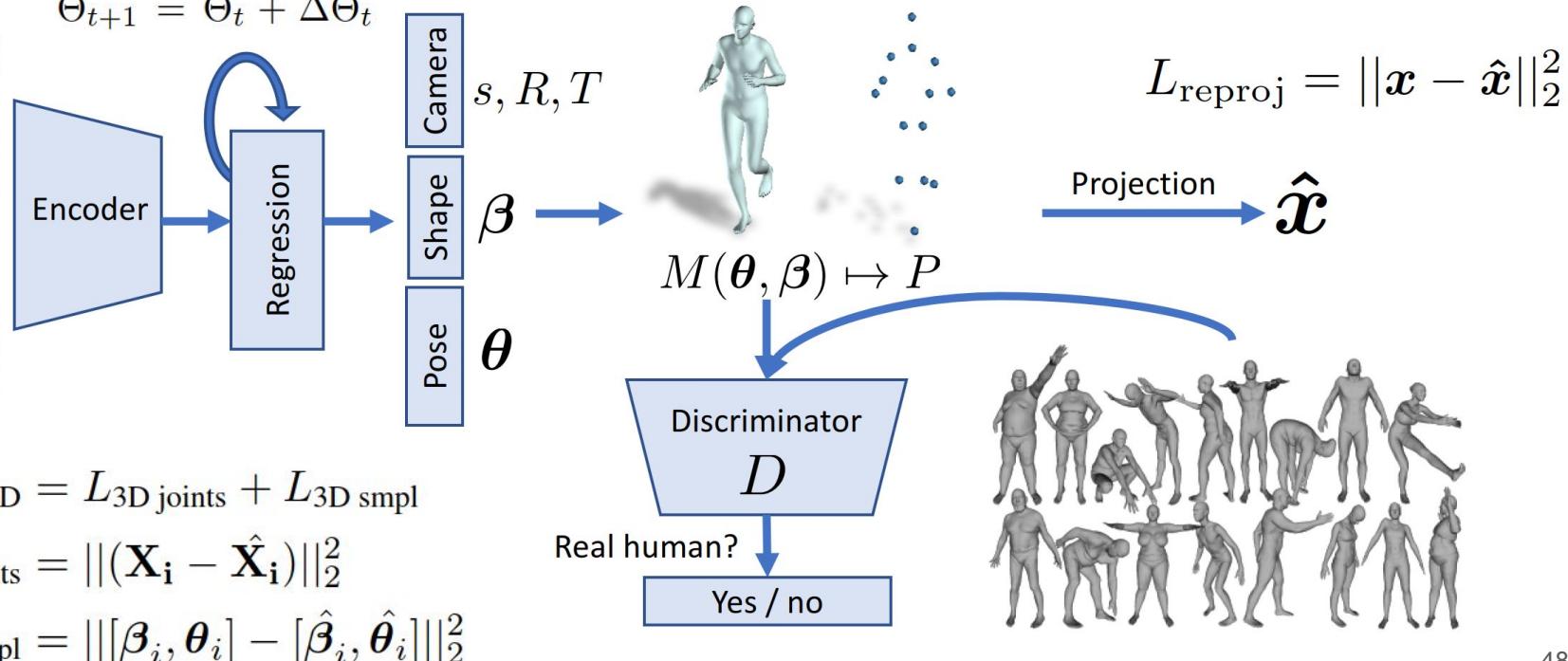
3D pose and shape

Pose and shape estimation with 3D supervision

Iterative 3D supervision with feedback



$$\Theta_{t+1} = \Theta_t + \Delta\Theta_t$$



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

$$L_{\text{joints}} = \|(\mathbf{X}_i - \hat{\mathbf{X}}_i)\|_2^2$$

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

Unsupervised pose and shape estimation

Method	MPJPE	Reconst. Error
Tome <i>et al.</i> [44]	88.39	
Rogez <i>et al.</i> [36]	87.7	71.6
VNect <i>et al.</i> [28]	80.5	
Pavlakos <i>et al.</i> [33]	71.9	51.23
Mehta <i>et al.</i> [27]	68.6	
Sun <i>et al.</i> [40]	59.1	
*Deep Kinematic Pose [52]	107.26	
*HMR	87.97	58.1
*HMR unpaired	106.84	67.45

- MPJPE: mean per joint position error,
- Reconstruction error: MPJPE after rigid alignment of the prediction with ground truth via Procrustes Analysis.

Table 2: **Human3.6M, Protocol 1.** MPJPE and reconstruction loss in mm. * indicates methods that output more than 3D joints.

Pros: Unsupervised approach is useful for in-the-wild applications.

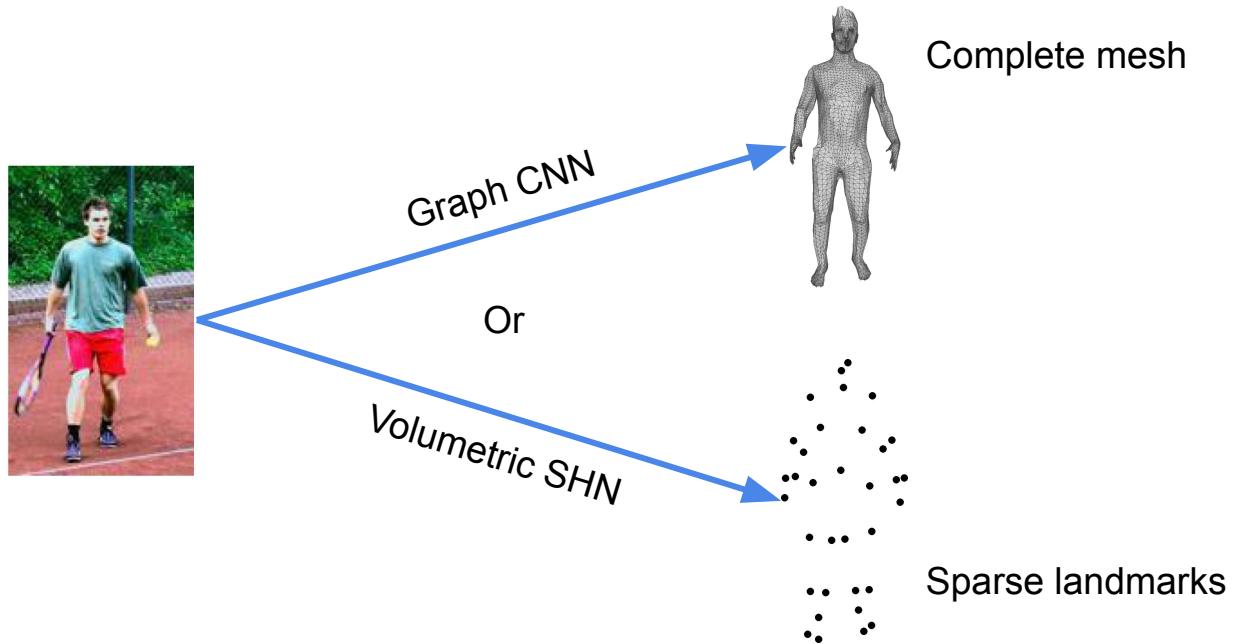
Cons: As expected, direct regression of SMPL parameters is challenging, even with 3D supervision.

2D pose estimation

3D pose estimation

3D pose and shape

Regressing body vertices rather than SMPL param.

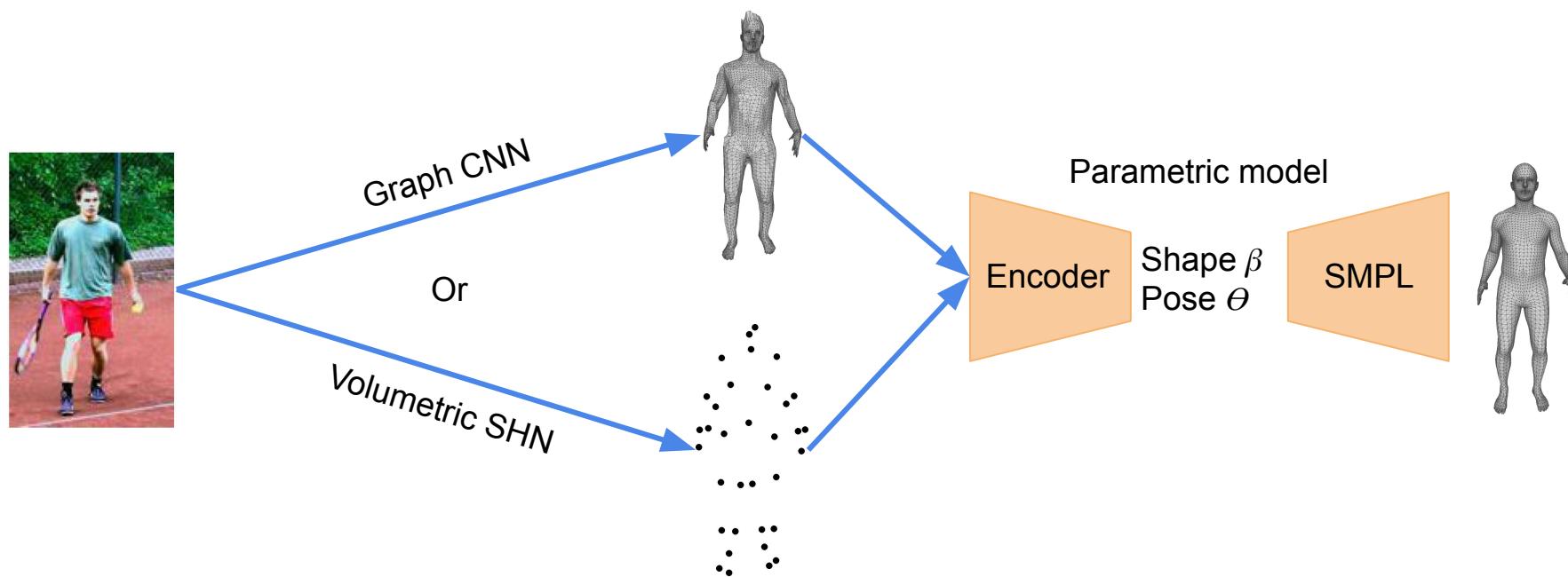


2D pose estimation

3D pose estimation

3D pose and shape

Regressing SMPL parameters on top of the 3D mesh/landmarks

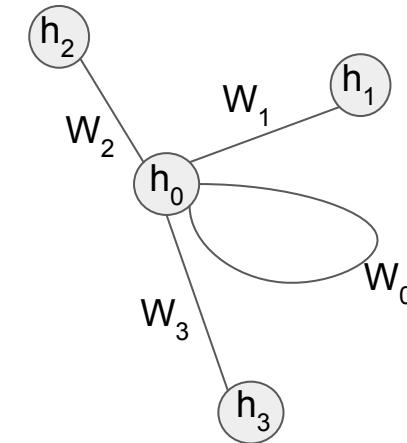


A review on graph convolutional network

$$f(H^{(l)}, A) = \sigma \left(\underbrace{D^{-\frac{1}{2}} A D^{-\frac{1}{2}}}_{\text{Symmetrically normalized adjacency matrix}} H^{(l)} W^{(l)} \right)$$

↓ Activation function ↓ Features ↓ Weights

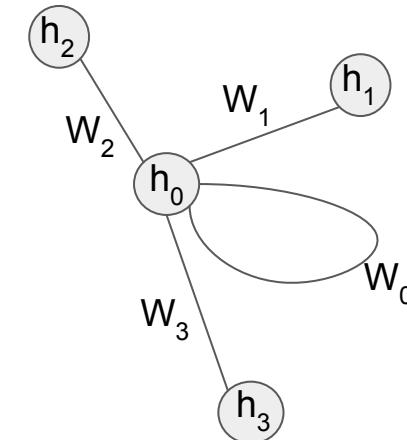
↓ Symmetrically normalized adjacency matrix



A review on graph convolutional network

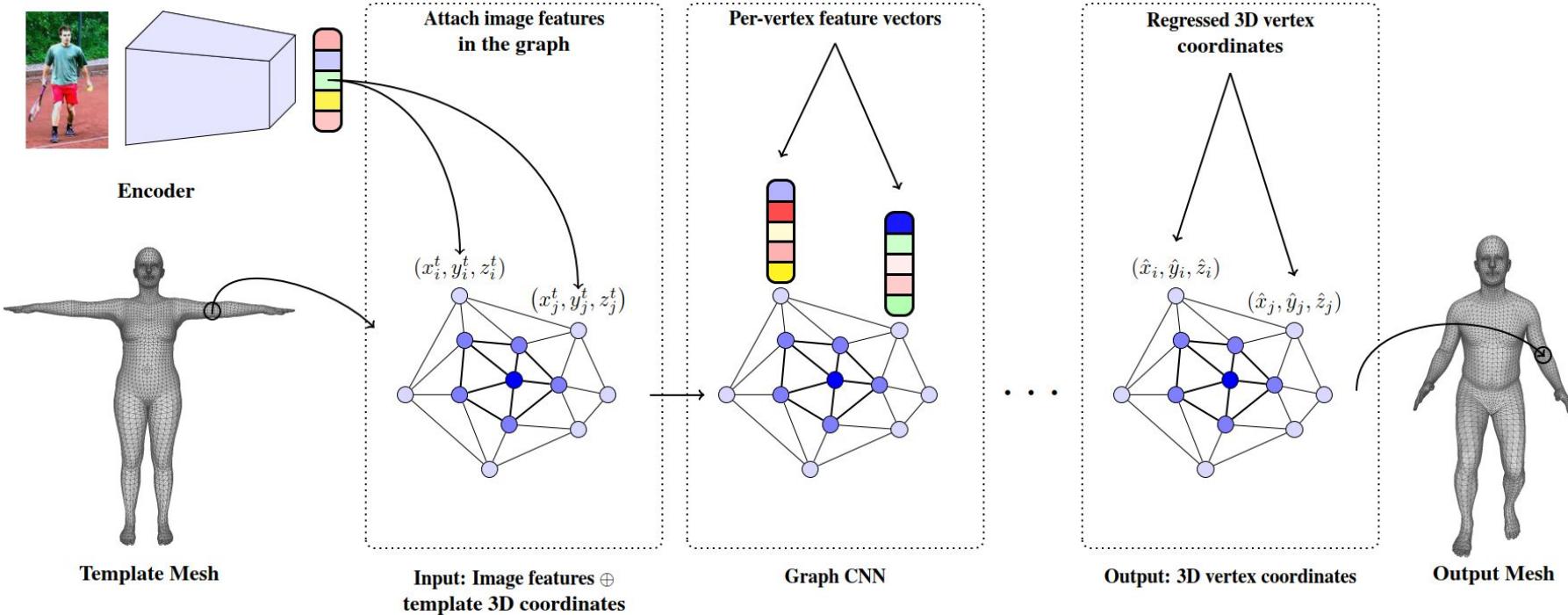
$$f(H^{(l)}, A) = \sigma \left(\underbrace{D^{-\frac{1}{2}} A D^{-\frac{1}{2}}}_{\text{Symmetrically normalized adjacency matrix}} H^{(l)} W^{(l)} \right)$$

Activation function Features Weights



- Each node has a different topology, therefore weights are shared among adjacent nodes, i.e. $\{W_0, W_n = W_1 | n > 0\}$,
- Arbitrary levels of adjacency can be applied,
- Batch normalization, dropout, pooling, etc are applicable,
- For simplicity, graph topology is fixed among all the samples,
- Due to weight sharing, GCN requires more filters in each layer than CNN.

Mesh regression through graph CNN

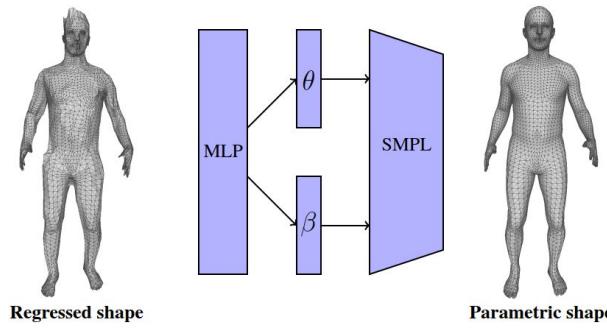


2D pose estimation

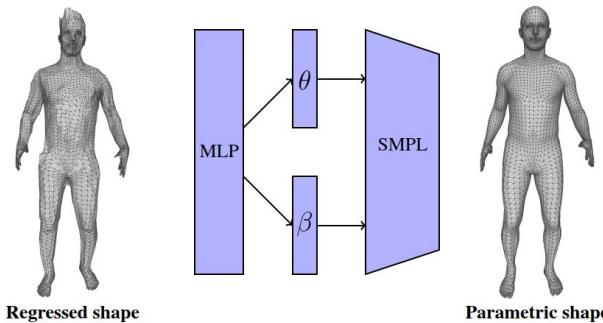
3D pose estimation

3D pose and shape

Mesh regression through graph CNN



Mesh regression through graph CNN



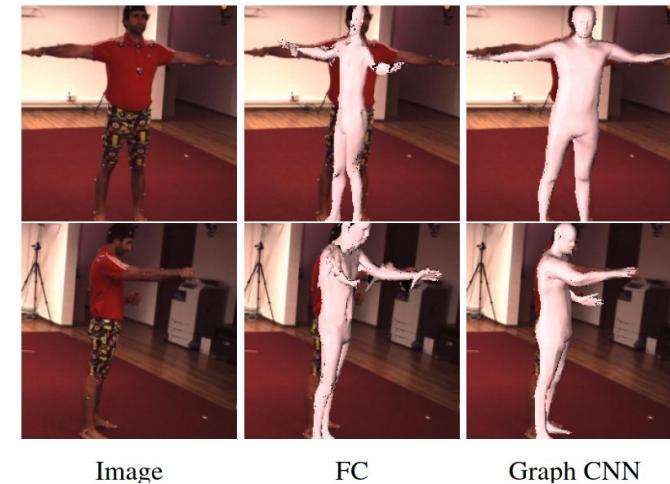
Method	MPJPE	Reconst. Error
SMPL Parameter Regression [15]	-	77.6
Mesh Regression (FC)	200.8	105.8
Mesh Regression (Graph)	102.1	69.0
Mesh Regression (Graph + SMPL)	113.2	61.3

Pros:

- Local convolutions over the mesh better deals with local minima than FC layers,
- Mesh regression is an easier problem than SMPL parameters regression.

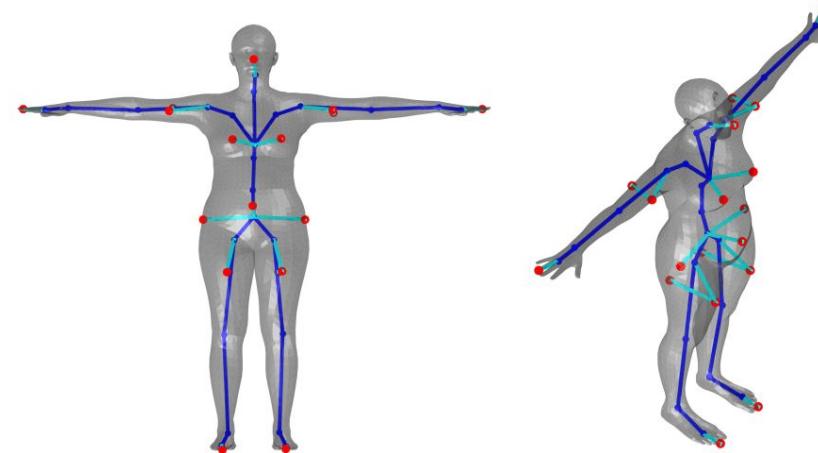
Cons:

- It easily overfits on the training set,
- It requires 3D ground truth hard to annotate in the real world applications.



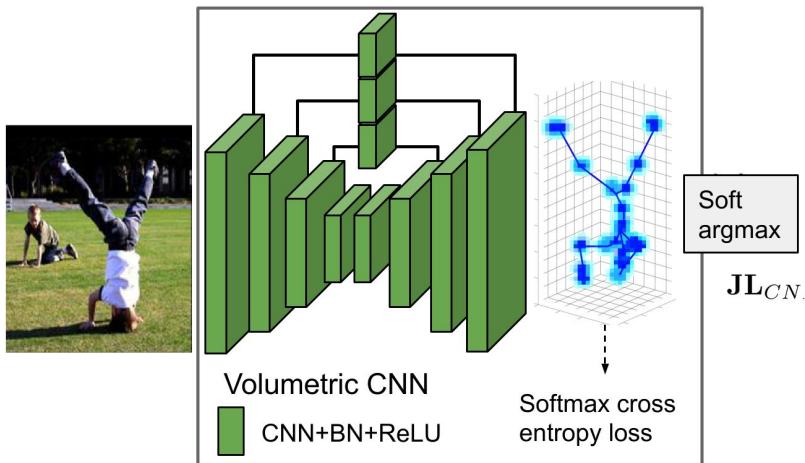
Sparse landmarks regression

- Landmarks can represent body surface geometry,
- They can be annotated easier than the whole mesh,
- They must be sparse but optimized in terms of the amount and location.



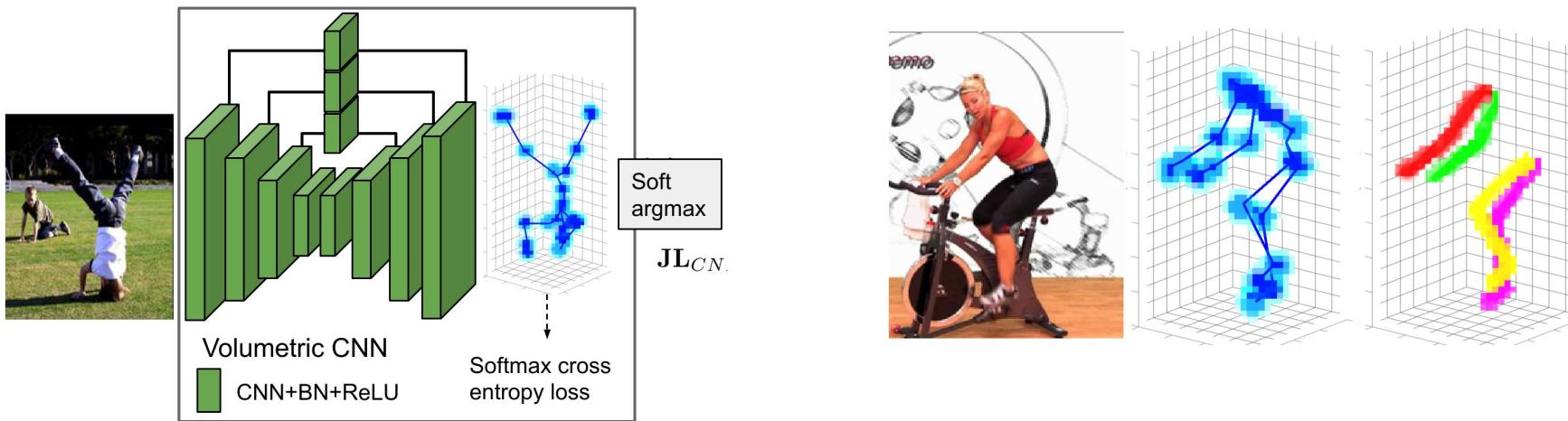
Joints + sparse landmarks regression through SHN

- The output layer of SHN is updated to be a volumetric tensor (width, height, depth, #joints + #landmarks),
- Volumetric SHN is a heavy model consuming a lot of memory in training,

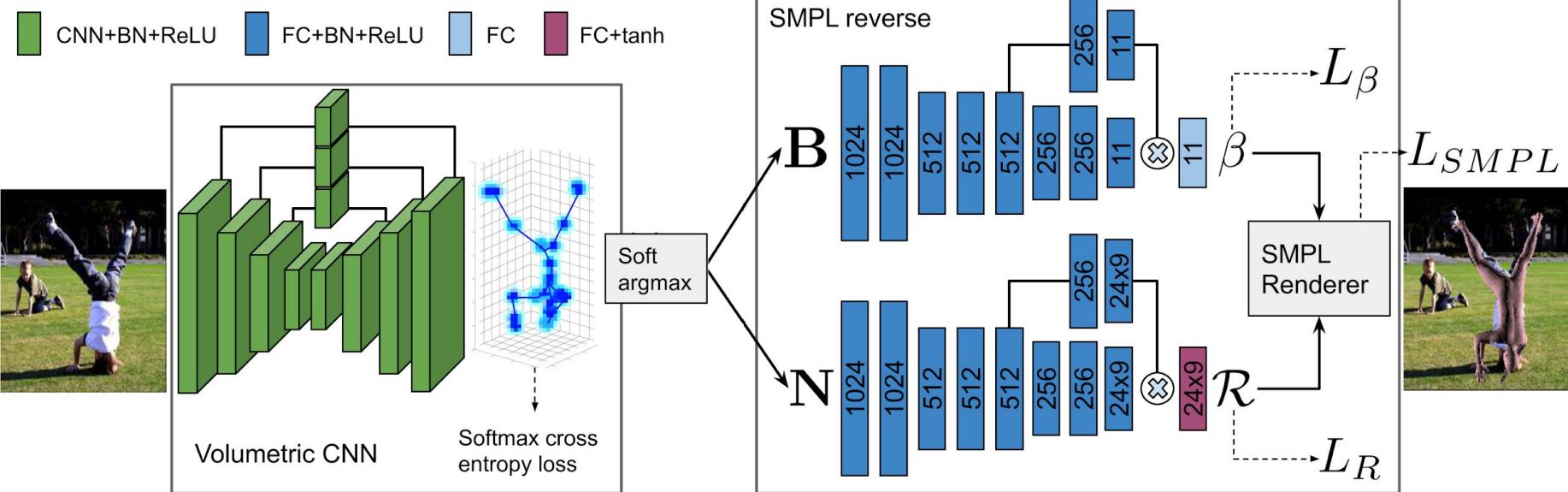


Joints + sparse landmarks regression through SHN

- The output layer of SHN is updated to be a volumetric tensor (width, height, depth, #joints + #landmarks),
- Volumetric SHN is a heavy model consuming a lot of memory in training,
- Inspired from compositional SHN, volumetric SHN can be updated with limb heatmaps, i.e. (width, height, depth, #joints + #landmarks + #limbs)



Regressing SMPL param. from joints and landmarks

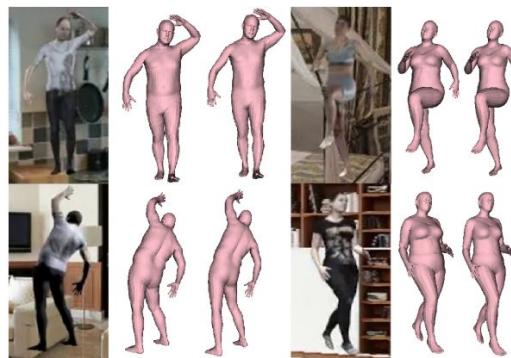


Joint and landmark locations are decomposed into B and N features:

- B: relative length w.r.t. the kinematic tree,
- N: relative norm w.r.t. the kinematic tree.

Regressing SMPL param. from joints and landmarks

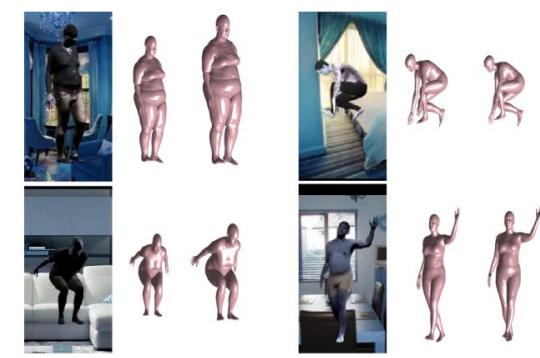
RGB Before After



Training SMPL regression with landmarks improves the estimated shape.



Training SHN end-to-end with SMPL regression improves SHN predictions.



Procrustes alignment of SMPL output with SHN output improves gender failures and global orientation.

Regressing SMPL param. from joints and landmarks

	Prot. 1	Prot. 2
Bogo [7]	-	82.3
Lassner [12]	-	80.7
Tung [31]*	98.4	-
Pavlakos [8]	-	75.9
Omran [13]	-	59.9
Kanazawa [9]	87.9	56.8
Kolotouros [35]	74.7	51.9
<i>ALL</i>	67.9	52.2
<i>ALL_{Proc}</i>	62.6	52.2

Methods in which the output is SMPL dense mesh

	Prot. 1	Prot. 2
Tome [25]	88.4	70.7
Pavlakos [27]	71.9	51.9
Zhou [36]	64.9	-
Martinez [20]	62.9	47.7
Sun [24]	59.1	48.3
Sun [37]**	64.1	-
Fang [38]	60.4	45.7
<i>SHN^{final}</i>	61.3	50.1
<i>SHN_{e2e}^{final}</i>	56.5	46.3

Methods in which the output is just 3D joints

Prot. 1: MPJPE
Prot. 2: reconstruction error

Kanazawa [9]: direct pose and shape regression

Kolotouros [35]: dense mesh regression by graph CNN

ALL: the whole pipeline

ALL_{Proc}: the results after procrustes alignment

SHN^{final}: SHN training with data augmentation

SHN_{e2e}^{final}: SHN results after end to end training

2D pose estimation

3D pose estimation

3D pose and shape

3D hand pose
estimation

3D hand pose estimation based on depth images

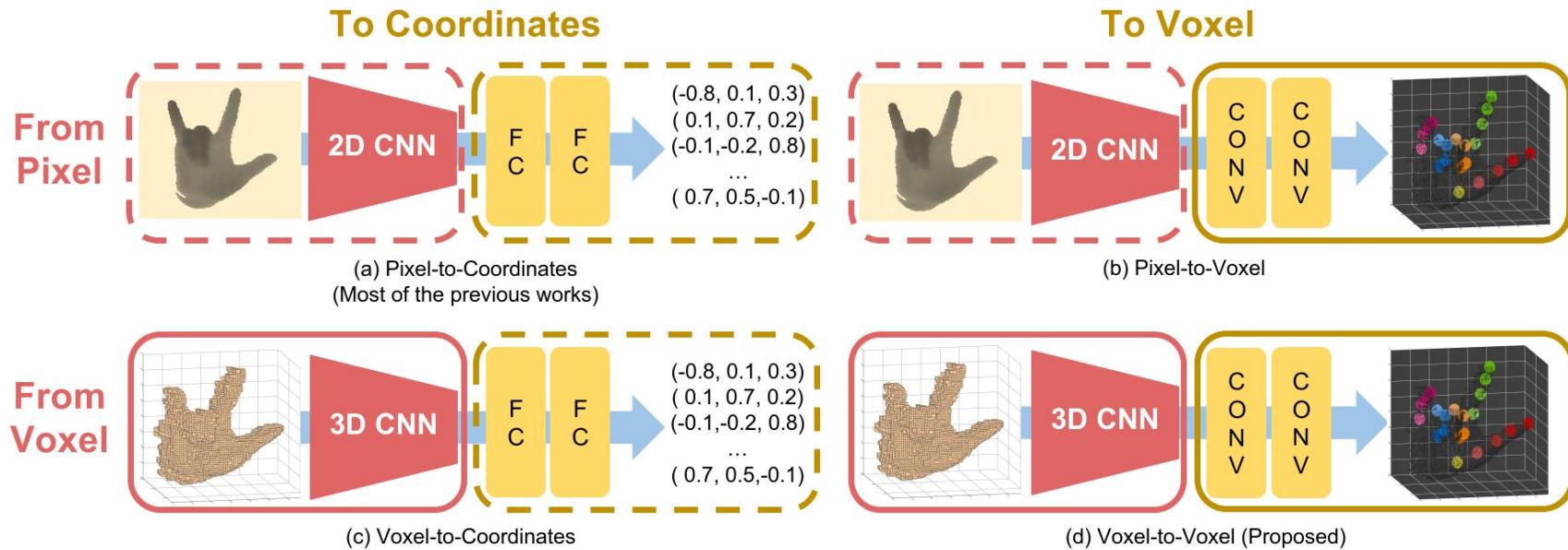




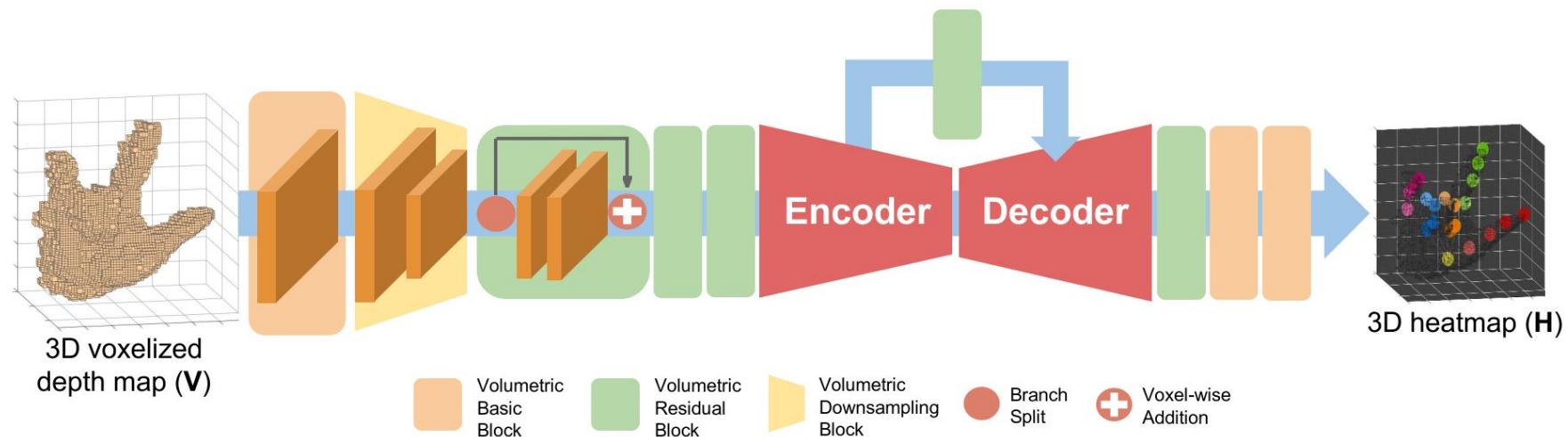
Applications

- Human-computer interaction,
- Virtual reality,
- Training robot hands,
- Sign language and gesture recognition.

Voxel to voxel predictions



Voxel to voxel predictions



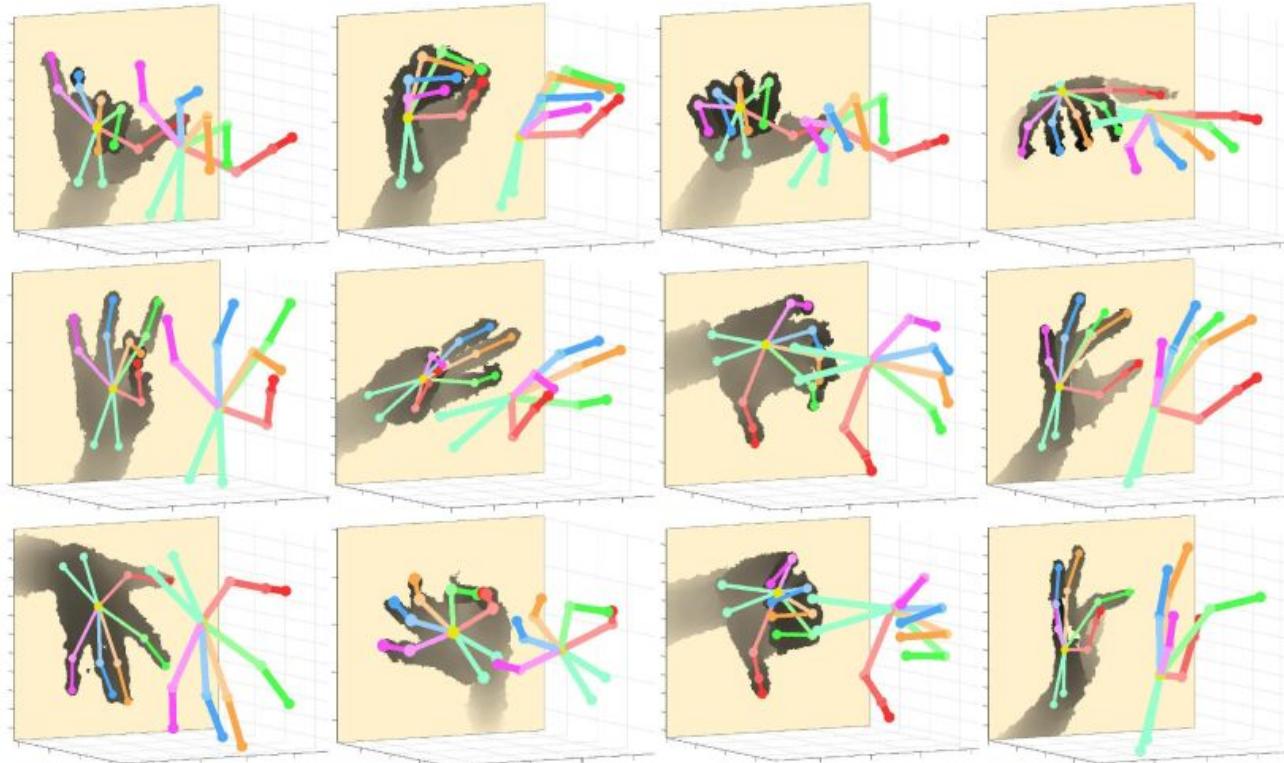
2D pose estimation

3D pose estimation

3D pose and shape

3D hand pose
estimation

Voxel to voxel predictions - results on NYU dataset





Further reading

- Pose guided region ensemble: image.ee.tsinghua.edu.cn/pdf/2018_chenxh_neucom.pdf
- Global-local pose: <https://arxiv.org/abs/1705.09606>
- Feature mapping: <https://arxiv.org/abs/1712.03904>
- Pose from RGB image: <https://arxiv.org/pdf/1705.01389.pdf>