Supplementary: Self-Supervised 3D Human Pose Estimation Via Part Guided Novel Image Synthesis

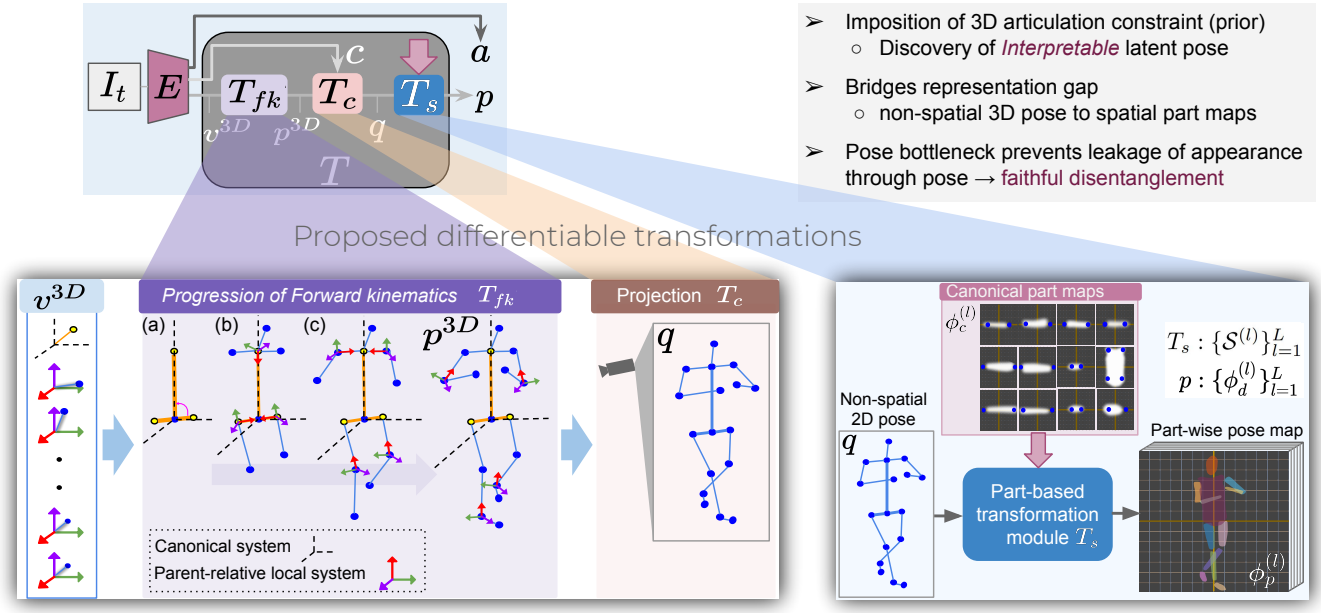


Figure 1: An overview of the full Encoder with details of the corresponding individual differentiable transformations.

This supplementary material is organized as follows:

- Sec. 1: Differentiable part transformation, T_s
- Sec. 2: Transforming local vectors to canonical pose
- Sec. 3: Self-supervised training procedure
  Fig. 4, 5, and 6 summarize the challenges and the proposed solutions for each self-supervised objective used in our learning framework.
- Sec. 4: Details of the encoder-decoder architecture
- Sec. 5: Other implementation details
- Sec. 6: Additional qualitative results

1. Differentiable part transformation, T_s

The part transformation module takes a set of 2D pose joint locations q and outputs a spatial part-wise pose map using a fixed dictionary of canonical part maps as shown in Fig 1, right panel. Each canonical part, $\phi_c^{(l)}$ is represented by 2 anchored joints, $r^{l(j_1)}$ and $r^{l(j_2)}$. Here, $l(j_1)$ denotes parent joint index of the limb l, whereas $l(j_2)$ denotes index of the child joint. Similarly, for a camera projected 2D pose $q \in \mathbb{R}^{2J}$, the spatial joint locations for the limb l are denoted by $q^{l(j_1)}$ for the parent and $q^{l(j_2)}$ for the child joint.

Note that, in the canonical part map, $\phi_c^{(l)}$ all the limbs or body parts are aligned along the positive X-axis, *i.e.* a vector directed from $r^{l(j_1)}$ to $r^{l(j_2)}$ makes an angle of $0°$ with respect to positive X-axis. Also, mid-point of the line-segment joining $r^{l(j_1)}$ to $r^{l(j_2)}$ aligns with the origin $(0, 0)$ while the corresponding spatial indices, $u \in [-H/2, H/2] \times [-W/2, W/2]$, where H and W are respectively height and width of the spatial map.

The rotation angle for the spatial transformation, $\mathcal{S}^{(l)}$ required to align the canonical part along the vector directed from $q^{l(j_1)}$ to $q^{l(j_2)}$ is computed as

$$\theta^{(l)} = -\text{atan2}((q_y^{l(j_2)} - q_y^{l(j_1)}), (q_x^{l(j_2)} - q_x^{l(j_1)}))$$

Here, $q_x^{l(j_1)}$ and $q_y^{l(j_1)}$ represent the X and Y component of the spatial joint location $q^{l(j_1)}$, and similarly for $q^{l(j_2)}$. Following this, a unidirectional scaling parameter along both X and Y axis is computed as

$$\gamma_x^{(l)} = dist(r^{l(j_1)}, r^{l(j_2)})/dist(q^{l(j_1)}, q^{l(j_2)}); \;\; \gamma_y^{(l)} = 1$$

Here, *dist* represents the Euclidean distance between the 2D spatial locations. We obtain the translation parameters, $(\mu_x^{(l)}, \mu_y^{(l)})$, as position of the mid-point between the joint locations, $(q_x^{l(j_1)}, q_y^{l(j_1)})$ and $(q_x^{l(j_2)}, q_y^{l(j_2)})$. Finally, the spatial transformation operation $\mathcal{S}^{(l)}$ is parameterized by $\theta^{(l)}$, $(\gamma_x^{(l)}, \gamma_y^{(l)})$, and $(\mu_x^{(l)}, \mu_y^{(l)})$, which are indirectly a function of $(q^{l(j_1)}, q^{l(j_2)}, r_c^{l(j_1)}, r_c^{l(j_2)})$. Following this, the canonical spatial indices $u : (u_x, u_y)$, are transformed to $u' : (u'_x, u'_y)$ using the following affine transformation,

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \gamma_x^{(l)} \cos\theta^{(l)} & -\gamma_y^{(l)} \sin\theta^{(l)} & -\mu_x^{(l)} \\ \gamma_x^{(l)} \sin\theta^{(l)} & \gamma_y^{(l)} \cos\theta^{(l)} & -\mu_y^{(l)} \end{bmatrix} * \begin{bmatrix} u'_x \\ u'_y \\ 1 \end{bmatrix}$$

The above spatial index mapping of u' to u is proceeded by a differentiable image sampling [2] to obtain the final spatially transformed part maps, *i.e.* $\phi_p^{(l)} = \mathcal{S}^{(l)} \circ \phi_c^{(l)}$.

Note that, for the torso part, which is represented by four anchored joints, a line segment through the mid-point of upper left, upper right anchor joints and the mid-point of lower left, lower right anchor joints is used to compute the relative rotation $\theta^{(l)}$. For torso, we allow scaling across both X and Y as opposed to the other limbs, where scaling is performed only along X.

2. Transforming local vectors to canonical pose

The primary neural output of the encoder network includes, **a)** a neck-pelvis to hip-line (line segment connecting the two hip joints) angle (see Fig. 2(a)) represented by two neural activations, *i.e.* the sin and cos component (with *tanh* non-linearity). **b)** 13 three dimensional unit vectors for 13 limb joints, which are defined at their respective parent relative local coordinate system (*i.e.* parent joint as the origin with axis directions obtained via Gram-Schmidt orthogonalization of the parent-limb vector and the *face-vector*).

Here, p^{3D} for *pelvis* is always set at origin, *i.e.* $(0, 0, 0)$. And p^{3D} for *neck* is set at $(0, 0, \alpha^{(j \to Pa(j))})$, where $\alpha^{(j \to Pa(j))}$ is the fixed bone length of the line segment joining $Pa(j)$ to j with $Pa(j)$ being the parent joint of *neck*, *i.e.* the *pelvis*. Note that, $\alpha^{(j \to Pa(j))}$ for each joint j, except the root node (*pelvis* joint) is acquired from the prior knowledge of relative human bone-lengths as a part of the 3D articulation constraints.

Following this, p^{3D} for *left-hip* and *right-hip* is computed by rotating the hip-line about the positive X-axis on the YZ-plane. Note that, middle point of the hip-line aligns with the origin.

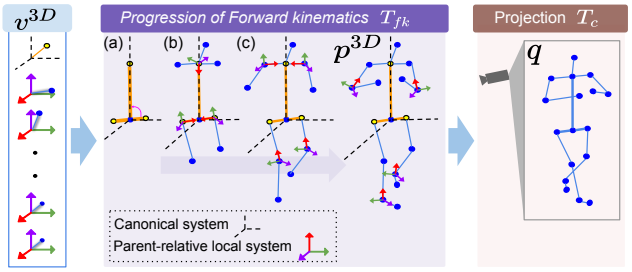


Figure 2: An overview of the proposed forward kinematic transformation to obtain a canonically aligned 3D pose p^{3D} from a set of parent-relative local pose vectors.

After obtaining p^{3D} for *pelvis*, *neck*, *left-hip* and *right-hip*, p^{3D} for rest of the joints is computed using a recursive forward kinematic formulation. For each joint j (excluding pelvis, neck and hip joints), the local coordinate system is defined by three mutually perpendicular directions, $a^{(j)}$, b, and $n^{(j)}$. Here, $a^{(j)}$ is a unit vector along the parent joint, *i.e.* along the line connecting $v^{3D}(j)$ to $v^{3D}(Pa(j))$. b is the *face-vector*, *i.e.* a unit vector along the positive X-axis and, n^j is the unit perpendicular vector obtained by performing cross product of $a^{(j)}$ with b. As discussed above, the raw neural values for each joint j are represented as $(v_x^{3D}(j), v_y^{3D}(j), v_z^{3D}(j))$ (*i.e.* *tanh* non-linearity followed by unit-vector normalization). The resultant direction in the canonical coordinate system is computed as,

$$\tilde{g}^{(j)} = v_x^{3D}(j)a^{(j)} + v_y^{3D}(j)b + v_z^{3D}(j)n^{(j)}$$

Here, $v_x^{3D}(j)$ is a scalar, whereas $a^{(j)}$ is a 3D vector in the Canonical coordinate system. A unit vector normalization of $\tilde{g}^{(j)}$ is represented as $g^{(j)}$. Then the final recursive forward kinematic formulation is implemented as,

$$p^{3D}(j) = p^{3D}(Pa(j)) + \alpha^{(Pa(j) \to j)} g^{(j)}$$

Here, $p^{3D}(j)$ is the position vector of joint j in the 3D canonical coordinate system. Finally, $p^{3D} : \{p^{3D}(j)\}_{j=1}^{J}$.

3. Training algorithm

As discussed in the main paper, the overall training procedure (see Algorithm 1) includes two training stages, a) optimization of the three consistency objectives, $\mathcal{L}_I^u$, $\mathcal{L}_I^c$, and $\mathcal{L}_{seg}$ and b) adaptation via decoupled energy minimization, which includes minimization of two energy functions $\mathcal{L}_{p_z^{3D}}$ and $\mathcal{L}_{a_s}$. We empirically validated the effectiveness of the proposed energy-based adaptation procedure by removing this step from the self-supervised training algorithm, denoted as *Ours(unsup) w/o adaptation* in Table 1.

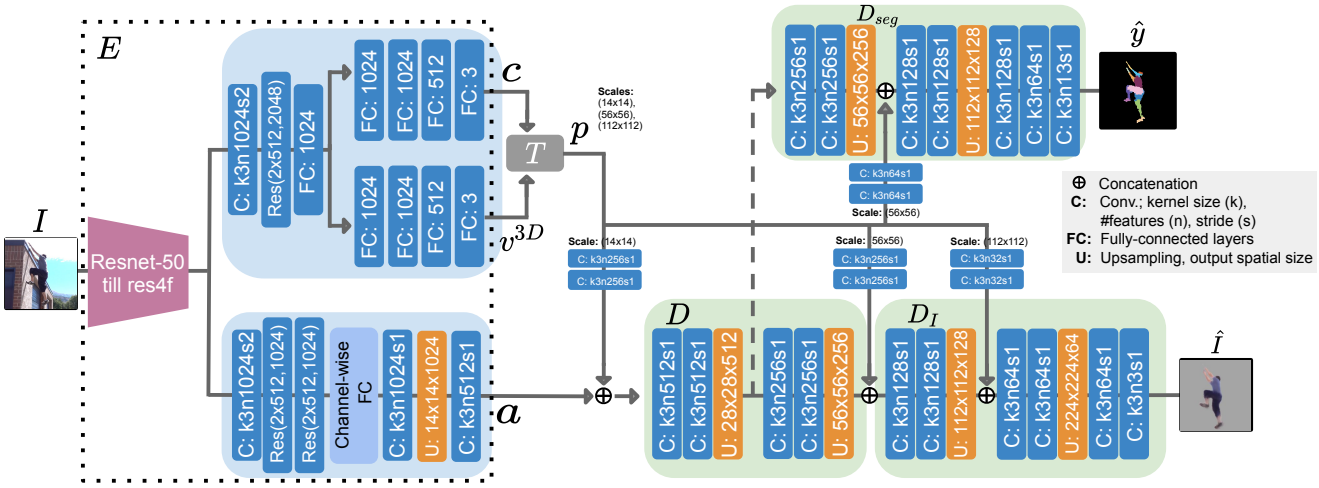


Figure 3: Architecture details of the the encoder-decoder model. Here, the *Channel-wise FC* (fully-connected) layer is inspired from [3], which is employed to allow interactions among the extreme spatial locations to account for diverse part deformations. Note that, the appearance representation a is expected to be generic across all the frames depicting the same person appearance (*i.e.* irrespective of the pose variations). Here, $Res(.,.,.)$ denotes a residual block as used in *Resnet50*.

/*Initialization of parameters */
θ_E: Trainable parameters of the Encoder E
θ_D: Trainable parameters of the Decoder (includes D, D_I, and D_{seg})
for *iter* $<$ *MaxIter* **do**
  /* Decoupled energy minimization. */
  if *iter* $\pmod 2 \neq 0$ **then**
    Update θ_E by optimizing $\mathcal{L}_{p_z^{3D}}$ and $\mathcal{L}_{a_s}$ in separate *Adagrad* optimizers on frozen θ_D.
  else
    Update θ_D by optimizing $\mathcal{L}_{p_z^{3D}}$ and $\mathcal{L}_{a_s}$ in separate *Adagrad* optimizers on frozen θ_E.
  end
  /* Optimize the consistency objectives. */
  Update (θ_E, θ_D) by optimizing $\mathcal{L}_I^u$, $\mathcal{L}_I^c$, and $\mathcal{L}_{seg}$ in separate *Adagrad* optimizers.
end

Algorithm 1: Training algorithm with the proposed adaptation via decoupled energy minimization.

The proposed energy minimization procedure to match the predicted 2D pose with the true 2D pose distribution is motivated from energy-based Generative Adversarial Networks [5]. Here, the decoder parameters are updated to realize a faithful $\hat{I}_{a_s}^{p_z}$ (*i.e.* natural looking), as the frozen encoder expects $\hat{I}_{a_s}^{p_z}$ to match its input distribution of real images (*i.e.* I_s or I_t) for an effective energy minimization (*i.e.* the pose and appearance extraction). Here, the encoder can be perceived as a frozen energy network as used in energy-based GAN [5]. A similar analogy applies while updating the encoder parameters with gradients from the frozen decoder. Each alternate energy minimization step is preceded by an overall optimization of the above consistency objectives, where both encoder and decoder parameters are updated simultaneously as shown in Algorithm 1.

Table 1: Ablation analysis, highlighting importance of various constraints and regularization in the proposed self-supervised 3D pose estimation framework.

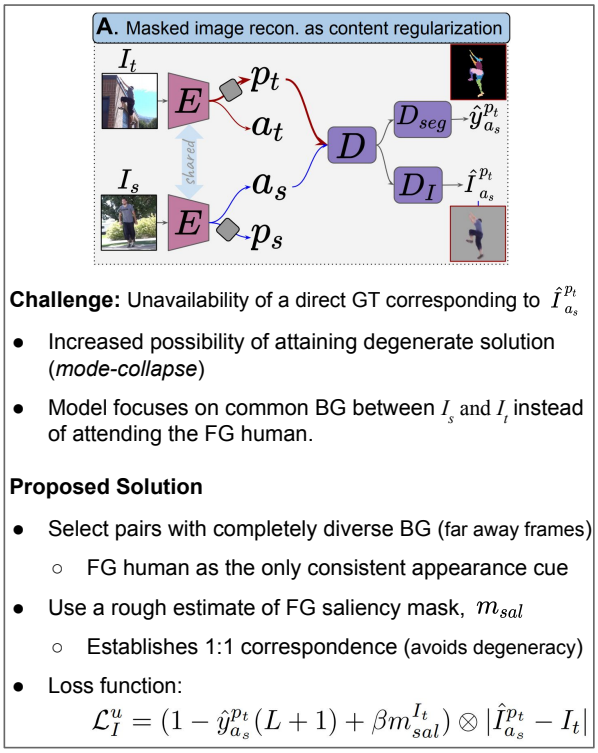
Method (unsup.)	MPJPE(↓) on H36M	3DPCK(↑) on MPI-3DHP
Ours(unsup) w/o $T_{fk} \circ T_c$	126.8	51.7
Ours(unsup) w/o m_{sal}	189.4	35.7
Ours(unsup) w/o adaptation	123.7	54.6
Ours(unsup)	99.2	77.4

Figure 4, 5, and 6 clearly list the challenges and the proposed solutions for each self-supervised objective.

4. Architecture

The architecture is an encoder-decoder setup where the encoder E encodes the pose and appearance information from an input image I. The decoder takes the concatenated representation of the FG appearance, a and pose, p as input to obtain two output maps, i) a reconstructed image $\hat{I}$, and ii) a predicted part segmentation map $\hat{y}$ via a bifurcated CNN decoder. The common decoder branch, D consists of a series of up-convolutional layers conditioned on the spatial pose map p at intermediate layer inputs (*i.e.* multi-scale pose conditioning). D_I and D_{seg} follow up-convolutional

layers to their respective outputs. Figure 3 shows the detailed architecture.



Challenge: Unavailability of a direct GT corresponding to $\hat{I}^{p_t}_{a_s}$

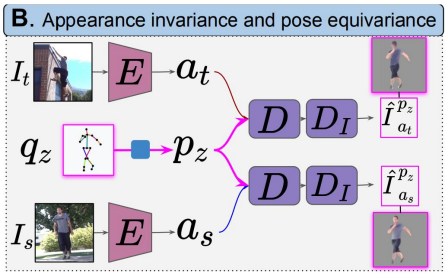
- Increased possibility of attaining degenerate solution (*mode-collapse*)
- Model focuses on common BG between I_s and I_t instead of attending the FG human.

Proposed Solution

- Select pairs with completely diverse BG (far away frames)
  - FG human as the only consistent appearance cue
- Use a rough estimate of FG saliency mask, m_{sal}
  - Establishes 1:1 correspondence (avoids degeneracy)
- Loss function:

$$\mathcal{L}^u_I = (1 - \hat{y}^{p_t}_{a_s}(L+1) + \beta m^{I_t}_{sal}) \otimes |\hat{I}^{p_t}_{a_s} - I_t|$$

Figure 4: A summary of the first self-supervised objective.



Challenge: Both $\hat{y}^{p_t}_{a_s}$ and m_{sal} are unreliable

- $\hat{y}^{p_t}_{a_s}$ depends on an unreliable predicted pose p_t
- m_{sal} captures general visual saliency (not human-specific)

Proposed Solution

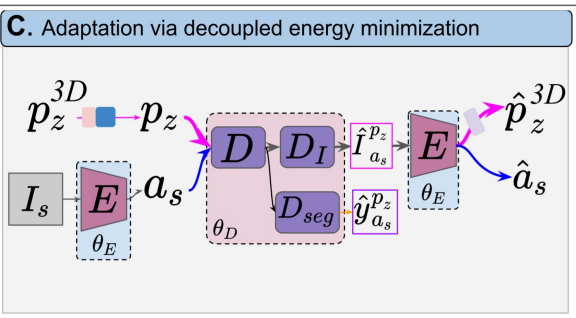
- Make use of reliable unpaired 2D pose data q_z
- Training objective inspired from simultaneous appearance invariance and pose equivariance

$$\mathcal{L}^c_I = w^{p_z}_{fg} \otimes |\hat{I}^{p_z}_{a_s} - \hat{I}^{p_z}_{a_t}| + (1 - w^{p_z}_{fg}) \otimes |\hat{I}^{p_z}_{a_s} - BG_c|$$

- Part segmentation on reliable 2D pose samples q_z

$$\mathcal{L}_{seg} = (1 - w_{unc}) \otimes CE(\hat{y}^{p_z}_{a_s}, y^{p_z}) + w_{unc} \otimes SE(\hat{y}^{p_z}_{a_s})$$

Figure 5: Summary of the second self-supervised objective.



Challenge: Model inculcates discrepancy between the predicted pose and the true pose distributions

- $\mathcal{L}^c_I$ and $\mathcal{L}_{seg}$ rely on true pose p^{3D}_z
- whereas, $\mathcal{L}^u_I$ relies on predicted pose q_t

Proposed Solution: Adaptation via energy minimization

- Two energy functions defined at the output of the secondary encoder via cyclic auto-encoding:

$$\mathcal{L}_{p^{3D}_z} = |p^{3D}_z - \hat{p}^{3D}_z| \qquad \mathcal{L}_{a_s} = |a_s - \hat{a}_s|$$

- We avoid a direct encoder decoder interaction during gradient back-propagation, by updating the encoder parameters, while freezing the decoder parameters and vice-versa.
- Inspired from energy-based GAN. Reutilization of encoder and decoder as energy function via decoupled parameter update
  - avoids the use of ad-hoc adversarial discriminator
  - simplified training regime

Figure 6: Summary of energy-based adaptation objective.

5. Other implementation details

While training, we use separate AdaGrad optimizers [1] for each loss term at alternate training iterations thereby avoiding manual loss balancing. The hyperparameter β, in the loss function $\mathcal{L}^u_I$ avoids the model from producing degenerate solutions (Fig. 5B, main paper, results without m_{sal}). However, considering unreliability of the saliency prediction algorithm, we reduce the strength of β after the model gains certain level of learning stability (*i.e.* after first 200k iterations). This helps to improve the model's ability to disentangle FG from the cluttered BG even beyond the unreliable m_{sal} predictions as a result of the pose dependent self-supervised consistency objectives. We use batch-size of 16 with an initial learning rate of 0.001 on a Tesla P100 machine (16GB VRAM). We train the model for $\sim$1600k learning iterations.

6. Additional results

Here we show additional qualitative results highlighting the effectiveness of the disentangled factors beyond the intended primary task of 3D pose estimation. We manipulate them to analyze their effect on the decoder synthesized

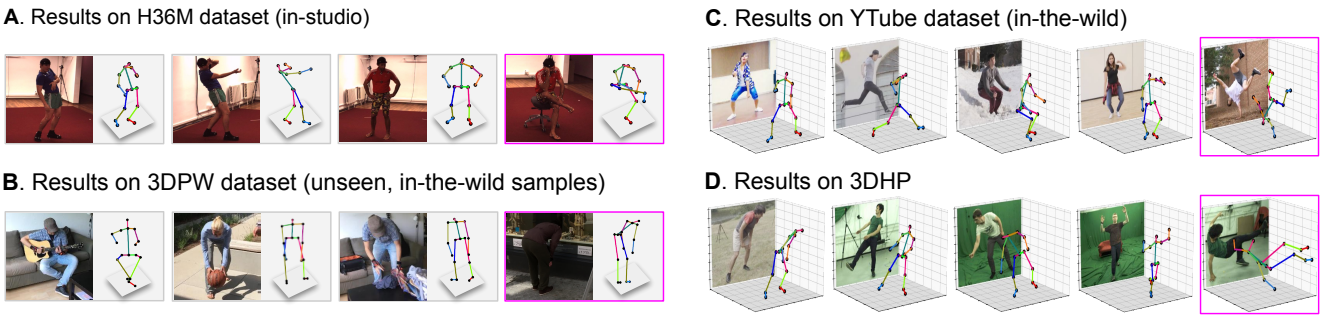


Figure 7: Qualitative results on 4 different datasets. Failure cases are highlighted in magenta which specifically occur in presence of multi-level inter-limb occlusion (see 3DPW failure case) and very rare, athletic poses (see YTube first failure case) . However, the model faithfully attends to single-level occlusions, enabled by the depth-aware part representation.

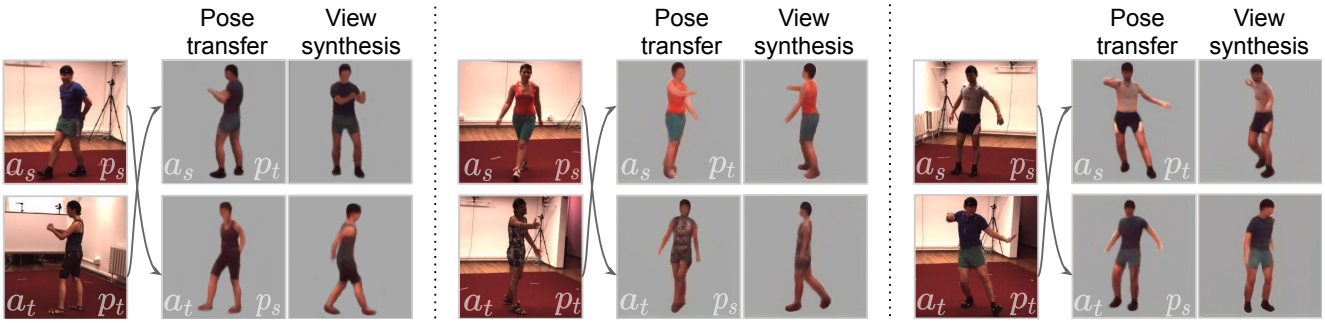


Figure 8: Novel image synthesis via latent manipulation of a, p and c on H3.6M dataset. It also shows the effect of independent non-rigid (pose transfer) and rigid (view synthesis) variations as a result of explicit disentanglement.

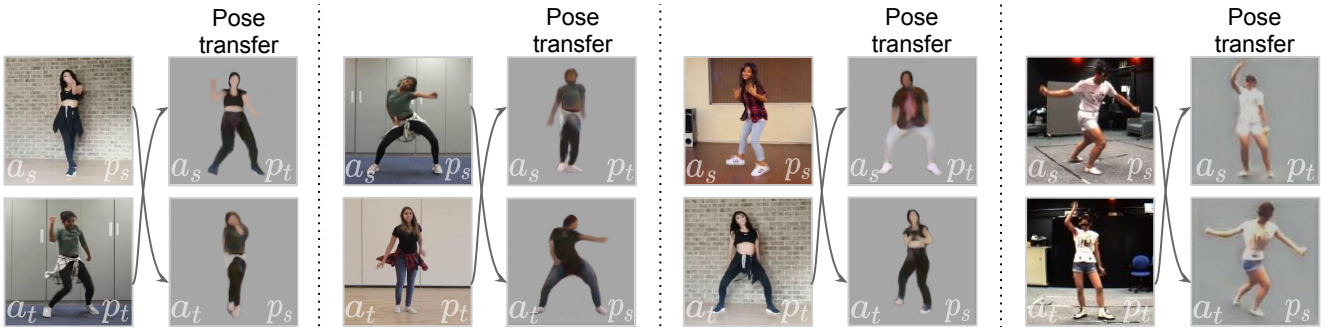


Figure 9: Pose transfer results on wild images from YTube dataset with diverse FG and BG appearance, and pose variations.

Table 2: Segmentation results on mean IOU (↑) for H3.6M.

Methods	Sup.	FG vs BG	FG Parts
SURREAL Synth. [4]	Full segmentation	46.35	42.91
SURREAL Real [4]	Full segmentation	49.61	46.32
Ours(weakly-sup)	Full-2D pose	47.86	44.24
Ours(unsup)	No supervision	45.54	42.36

output image. In Fig. 8 and Fig. 9, we show *pose transfer* results where the pose obtained from an image is transferred to the appearance of another. However, in *view synthesis*, we randomly vary the camera extrinsic values in c. The results shown are obtained from *Ours(unsup)* model, which is trained on the mixed YTube+H3.6 dataset. This confirms our superior disentanglement performance. Fig. 7 depicts qualitative results for the primary 3D pose estimation task using *Ours(weakly-sup)* model explained in main paper. In Fig. 7B, we show results on the unseen 3DPW dataset, where the model has not seen this dataset even during self-supervised training. A consistent performance on such unseen dataset further establishes generalizability of the proposed learning framework. We also show results on MPI-INF-3DHP and in-the-wild YTube dataset.

Quantitative comparison of part segmentation on Human3.6M is reported in Table 2. We achieve comparable

results against the prior arts, in absence of additional supervision as used in prior arts.

References

[1] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 4

[2] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015. 2

[3] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 3

[4] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 5

[5] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *ICLR*, 2017. 3