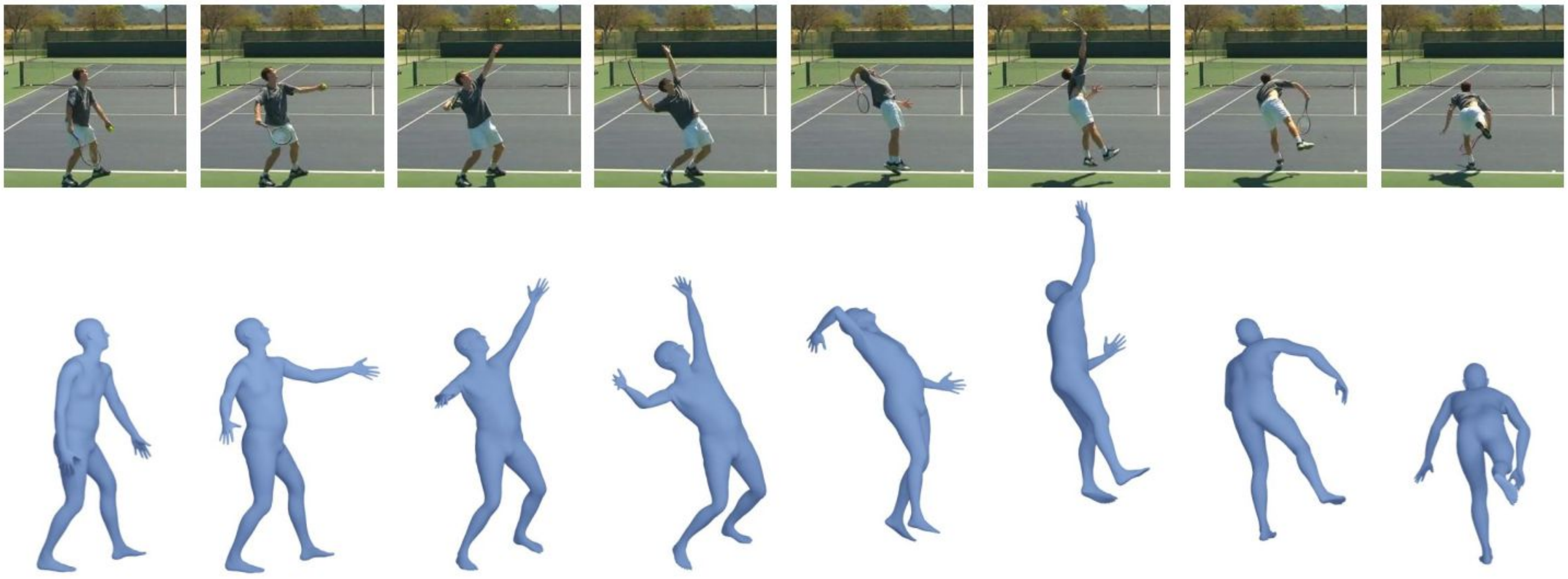
Uncertainty-Aware Human Mesh Recovery from Video by Learning **Part-Based 3D Dynamics**



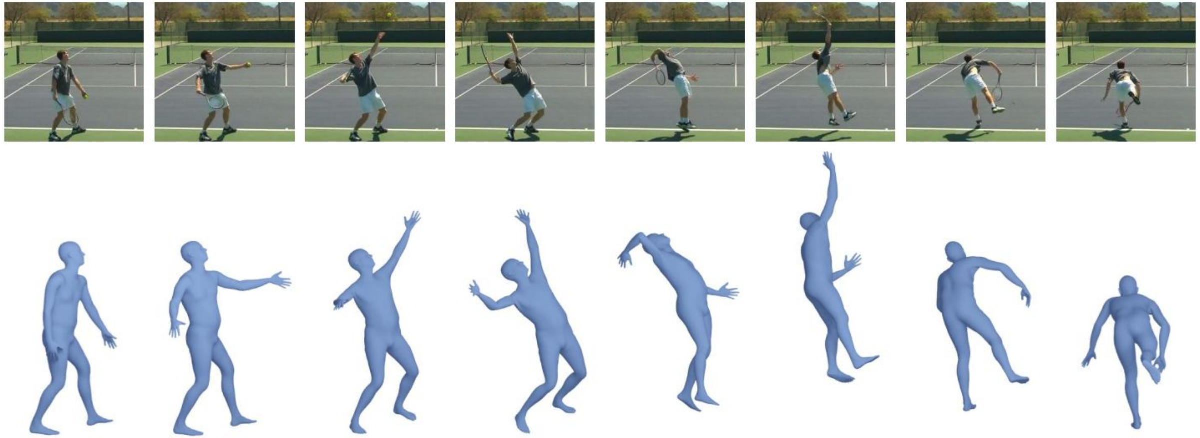
2D-to-3D

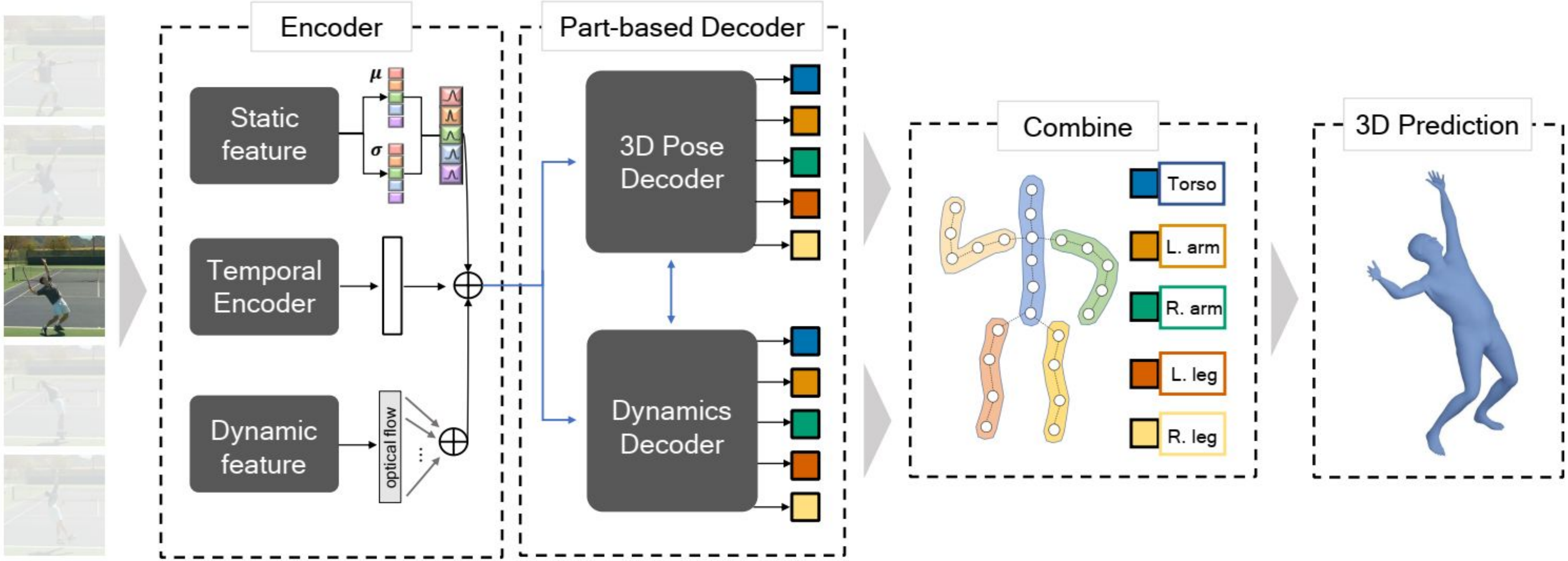
Problem Statement

➔ To recover temporally consistent human mesh from RGB video.

➔ Temporally consistent means :

- The changes on the surface of the mesh should be smooth across the frames.
- There should not be any extreme variations in the pose or body shape in between adjacent frames.




Encoder
Static feature
μ
σ
Temporal Encoder
Dynamic feature
optical flow
Part-based Decoder
3D Pose Decoder
Dynamics Decoder
Combine
Torso
L. arm
R. arm
L. leg
R. leg
3D Prediction

Formulation

- Given input video $V = \{ I_t \}$ containing T frames, where I_t denotes t^{th} frame.
- Goal is to predict human motion sequences $M = \{ \boldsymbol{\Theta_t} \}$ [t = 1 to T], where $\boldsymbol{\Theta_t}$ represents SMPL parameters for t^{th} frame.
- SMPL Parameters :
  - $\boldsymbol{\theta} \in R^{24 \times 3}$ → **Pose parameters** : Models global body rotation and relative rotation of 23 joints in axis-angle representation.
  - $\boldsymbol{\beta} \in R^{10}$ → **Shape parameters** : First 10 coefficients of shape space given by PCA
- Given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, SMPL defines a function $\mathbf{S}(\boldsymbol{\theta}, \boldsymbol{\beta}) \in R^{6890 \times 3}$ which outputs a 3D human mesh.

Uncertainty-Aware Temporal Feature

- Given a sequence of input frames $I_1 \ldots\ldots I_T$, a feature vector is extracted per frame using a pretrained ResNet : $f_1 \ldots\ldots f_{T,}$ where $f_t \epsilon R^{2048}$

- These features are passed to a GRU Layer that yields temporal features : $g_1 \ldots\ldots g_{T,}$ where $g_t \epsilon R^{2048}$

- G_t is then concatenated with two more features :

    - **Uncertainty-aware static feature**

    - **Dynamic feature**

Uncertainty-Aware Static Feature

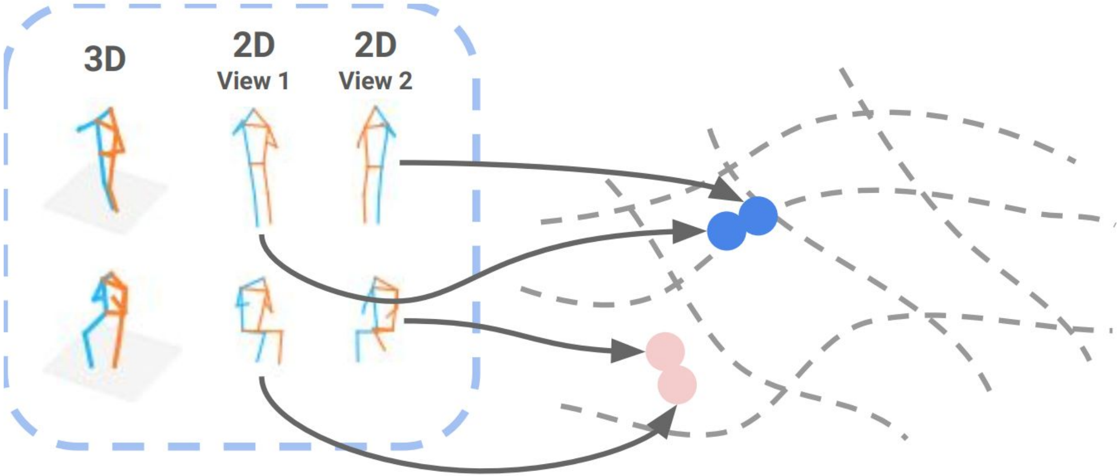
➔ **Choice for the feature :** An embedding vector z for 2D pose that should remain consistent across the views.

➔ This is a difficult task as various human pose in 3D space can be projected to same 2D pose which leads to an ambiguity.

➔ To solve this issue **Pr-VIPE (Probabilistic View Invariant Pose Embedding)** is used.

VIPE

➔ VIPE : View Invariant Pose Embedding.



VIPE : Approach

- ➔ Goal is to embed 2D poses such that their distances in the embedding space corresponds to similarities between corresponding 3D pose in Eucledian space.

- ➔ Two 3D poses are said to be matched if they are visually similar regardless of the viewpoint.

- ➔ Given two sets of 3D keypoints (y_i , y_j), matching indicator function is defined as :

$$m_{ij} = \begin{cases} 1, & \text{if NP-MPJPE}(\boldsymbol{y}_i, \boldsymbol{y}_j) \leqslant \kappa \\ 0, & \text{otherwise,} \end{cases}$$

- ➔ ***k*** controls visual similarity, and to quantify visual similarity, **N**ormalized **P**rocrustes-aligned **M**ean **P**er **J**oint **P**osition **E**rror is used.

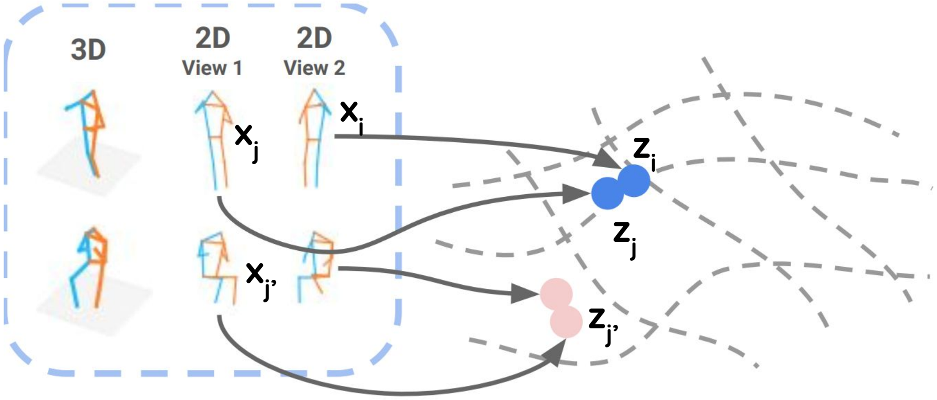
VIPE : Triplet Ratio Loss

➔ Let input 2D keypoints **x** ϵ **R**^{n} and output embedding vector **z** ϵ **R**^{d}.

➔ Goal is to learn a mapping function :

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^d, \text{ such that } D(\boldsymbol{z}_i, \boldsymbol{z}_j) < D(\boldsymbol{z}_i, \boldsymbol{z}_{j'}), \forall m_{ij} > m_{ij'}$$

➔ here, **z = f(x)** and **D(z_i , z_j)** is the distance measure in embedding space.



VIPE : Triplet Ratio Loss

➔ Let **p(m | x_i , x_j)** be the matching probability of two 3D poses y_i and y_j.

➔ If two 3D poses are identical then **p(m | x_i , x_j) = 1.**

➔ If two poses are sufficiently different, then **p(m | x_i , x_j)** should be small.

➔ For any given input triplet **(x_i , x_{i+} , x_{i-})** with **m_{ij+} > m_{ij-}** :

$$\frac{p(m|\boldsymbol{z}_i, \boldsymbol{z}_{i+})}{p(m|\boldsymbol{z}_i, \boldsymbol{z}_{i-})} \geqslant \beta,$$

➔ where ***β* > 1**. Applying negative logarithm both sides,

$$(-\log p(m|\boldsymbol{z}_i, \boldsymbol{z}_{i+})) - (-\log p(m|\boldsymbol{z}_i, \boldsymbol{z}_{i-})) \leqslant -\log \beta.$$

VIPE : Triplet Ratio Loss

➔ Now, for a batch-size **N,** triplet ratio loss can be defined as :

$$\mathcal{L}_{\text{ratio}} = \sum_{i=1}^{N} \max(0, D_m(\boldsymbol{z}_i, \boldsymbol{z}_{i+}) - D_m(\boldsymbol{z}_i, \boldsymbol{z}_{i-}) + \alpha)),$$

with distance kernel $D_m(\boldsymbol{z}_i, \boldsymbol{z}_j) = -\log p(m|\boldsymbol{z}_i, \boldsymbol{z}_j)$ and margin $\alpha = \log \beta$

➔ After learning the embedding, matching probability can be determined as :

$$p(m|\boldsymbol{z}_i, \boldsymbol{z}_j) = \sigma(-a||\boldsymbol{z}_i - \boldsymbol{z}_j||_2 + b),$$

where, **a** and **b** are learnable parameters.

VIPE : Positive Pairwise Loss

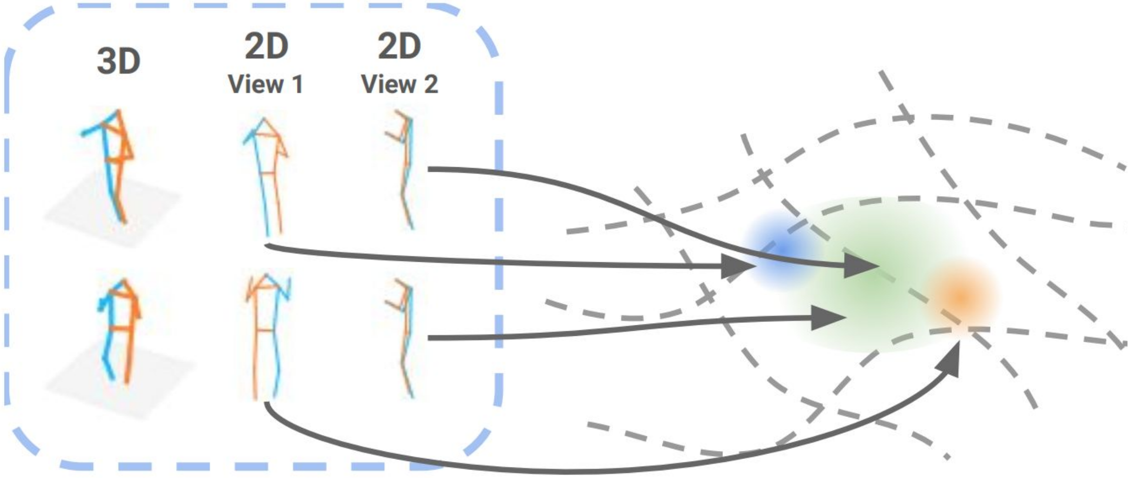
- ➔ To ensure identical 3D poses have higher matching probability, positive pairwise loss is also used.

$$\mathcal{L}_{\text{positive}} = \sum_{i=1}^{N} - \log p(m | \boldsymbol{z}_i, \boldsymbol{z}_{i+}).$$

- ➔ The combination of $\boldsymbol{L}_{ratio}$ and $\boldsymbol{L}_{positive}$ is used to train the embedding model.

Pr-VIPE

➔ Pr-VIPE : Probabilistic View Invariant Pose Embedding.



Pr-VIPE

- ➔ Let **p(m | x_i , x_j)** be the matching probability of two 3D poses y_i and y_j.
- ➔ Now instead of using indicator function, this probability is defined as:

$$p(m|\boldsymbol{x}_i, \boldsymbol{x}_j) = \int p(m|\boldsymbol{z}_i, \boldsymbol{z}_j) p(\boldsymbol{z}_i|\boldsymbol{x}_i) p(\boldsymbol{z}_j|\boldsymbol{x}_j) \mathrm{d}\boldsymbol{z}_i \mathrm{d}\boldsymbol{z}_j$$

- ➔ This can approximated by Monte Carlo Sampling, with ***K*** samples drawn from each distribution as :

$$p(m|\boldsymbol{x}_i, \boldsymbol{x}_j) \approx \frac{1}{K^2} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} p(m|\boldsymbol{z}_i^{(k_1)}, \boldsymbol{z}_j^{(k_2)}).$$

Pr-VIPE

- ➔ After learning the probabilistic embedding, the model outputs **mean** and **covariance** for a given set of 2D keypoints.

- ➔ Mean : $\boldsymbol{\mu}_t \in \mathbf{R}^{32}$ and Covariance : $\boldsymbol{\sigma}_t \in \mathbf{R}^{32}$ are then concatenated to form **uncertainty-aware static feature** $\mathbf{u}_t \in \mathbf{R}^{64}$

Recall : Uncertainty-Aware Temporal Feature

- ➔ Given a sequence of input frames $\mathbf{I_1 ..... I_T}$, a feature vector is extracted per frame using a pretrained ResNet : $\mathbf{f_1 ..... f_T}$, where $\mathbf{f_t \epsilon R^{2048}}$
- ➔ These features are passed to a GRU Layer that yields temporal features : $\mathbf{g_1 ..... g_T}$, where $\mathbf{g_t \epsilon R^{2048}}$
- ➔ $\mathbf{G_t}$ is then concatenated with two more features :
  - ◆ **Uncertainty-aware static feature** ☑
  - ◆ **Dynamic feature**

Dynamic Feature

➔ Optical flow is utilised here as it has strong cues for motion dynamics.

➔ For a frame $\mathbf{I}_t$ optical flow information is constructed by calculating homographies between successive frames within the interval $[\mathbf{I}_{t-15}, \mathbf{I}_t]$.

➔ 3x3 homography matrix is obtained by solving flow equations using SVD.

➔ All the homographies are stacked together to form dynamic feature $\mathbf{d}_t \in \mathbf{R}^{135}$

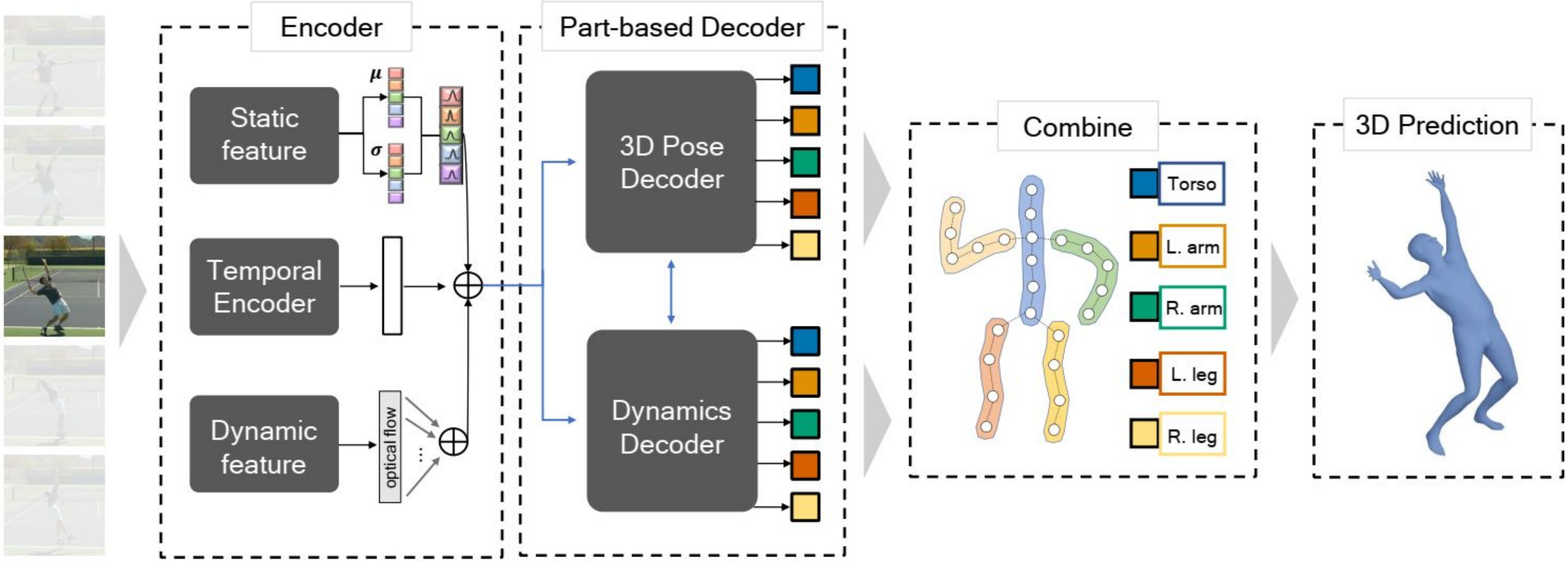
Recall : Uncertainty-Aware Temporal Feature

➔ Given a sequence of input frames $\mathbf{I_1 \ldots\ldots I_T}$, a feature vector is extracted per frame using a pretrained ResNet : $\mathbf{f_1 \ldots\ldots f_T}$, where $\mathbf{f_t \in R^{2048}}$

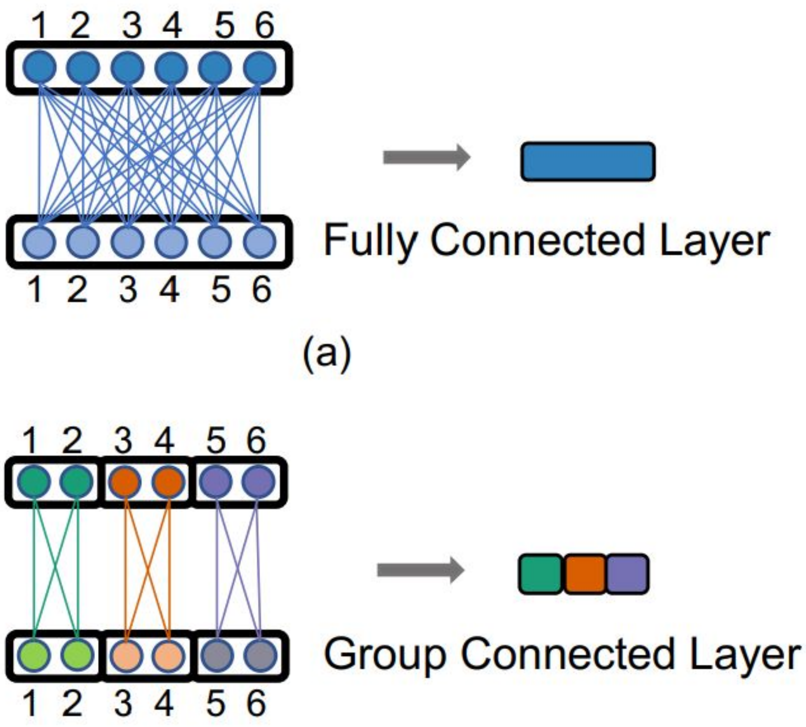
➔ These features are passed to a GRU Layer that yields temporal features : $\mathbf{g_1 \ldots\ldots g_T}$, where $\mathbf{g_t \in R^{2048}}$

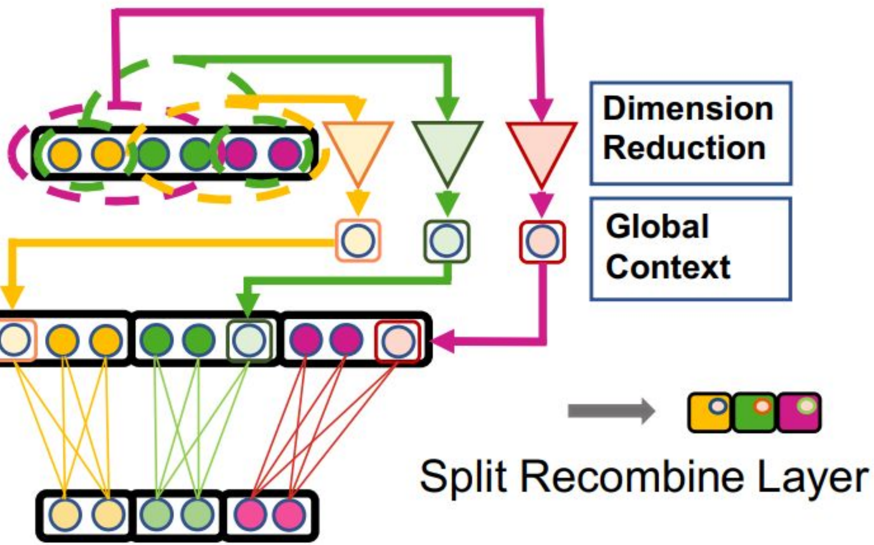
➔ $\mathbf{G_t}$ is then concatenated with two more features :

- **Uncertainty-aware static feature** ☑
- **Dynamic feature** ☑

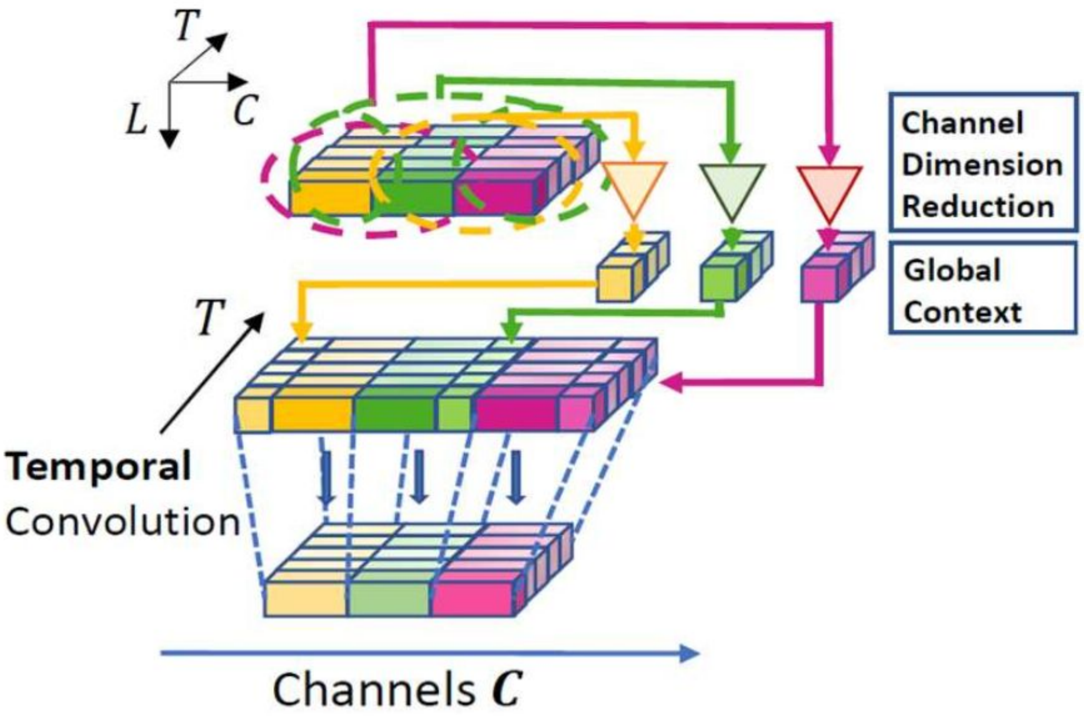

Encoder
Static feature
μ
σ
Temporal Encoder
Dynamic feature
optical flow
Part-based Decoder
3D Pose Decoder
Dynamics Decoder
Combine
Torso
L. arm
R. arm
L. leg
R. leg
3D Prediction

Split & Recombine





Split & Recombine : Temporal Convolution



Generator Loss Function

➔ Loss for the generator is proposed as:

$$L_{\mathcal{G}} = L_{2D} + L_{3D} + L_{SMPL},$$

$$L_{2D} = \sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2,$$

$$L_{3D} = \sum_{t=1}^{T} \|X_t - \hat{X}_t\|_2,$$

$$L_{SMPL} = \|\beta - \hat{\beta}\|_2 + \sum_{t=1}^{T} \|\theta_t - \hat{\theta}_t\|_2.$$

Evaluation

	Models	3DPW				MPI-INF-3DHP			Human3.6M		
		PA-MPJPE ↓	MPJPE ↓	MPVPE↓	Accel↓	PA-MPJPE↓	MPJPE↓	Accel↓	PA-MPJPE↓	MPJPE↓	Accel↓
Frame-based	Kanazawa *et al.* [3]	76.7	130.0	-	37.4	89.8	124.2	-	56.8	88	-
	Kolotouros *et al.* [29]	70.2	-	-	-	-	-	-	50.1	-	-
	Kolotouros *et al.* [30]	59.2	96.9	116.4	29.8	67.5	105.2	-	41.1	-	18.3
	Moon *et al.* [15]	57.7	93.2	110.1	30.9	-	-	-	41.1	**55.7**	13.4
	Choi *et al.* [18]	58.3	**88.9**	106.3	22.6	-	-	-	46.3	64.9	23.9
Temporal	Kanazawa *et al.* [2]	72.6	116.5	139.3	15.2	-	-	-	56.9	-	-
	Doersch *et al.* [5]	74.7	-	-	-	-	-	-	-	-	-
	Sun *et al.* [35]	69.5	-	-	-	-	-	-	42.4	59.1	-
	Kocabas *et al.* [27]	56.5	95.8	113.4	27.1	63.4	97.7	29.0	41.5	65.9	18.3
	Choi *et al.* [17]	55.8	95.0	111.5	7.0	62.8	97.4	**8.0**	41.1	62.3	**5.3**
	Ours	**52.2**	92.8	**106.1**	**6.8**	**59.4**	**93.5**	9.4	**38.4**	58.4	6.1

Results