2023 하계 세미나

Hand Pose Estimation

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

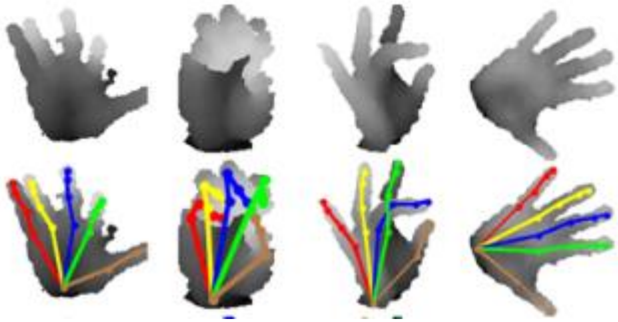
Presented By
Yeieun Hwang, 황예은

Contents

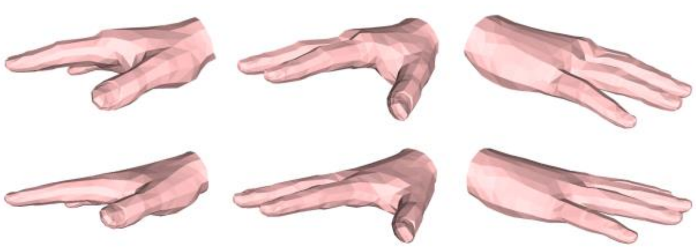
- Background
  - Hand Pose Estimation
- HTT: Hierarchical Temporal Transformer
  - Hierarchical Temporal Transformer for 3D Hand Pose Estimation and Action Recognition from egocentric RGB Videos (CVPR 2023)
- HaMuCo
  - HaMuCo: Hand Pose Estimation via Multiview Collaborative Self-Supervised Learning (ICCV 2023)

Background: Hand Pose Estimation

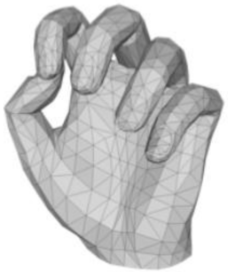
- Depth-based method
- RGB-based method
  - Skeleton-based method
    - Regressing hand joints directly
  - Model-based method
    - Using MANO, which can incorporate the hand prior and predict the hand mesh directly
  - Mesh-based method
    - Regressing each vertex directly with GCN, transformer or both



depth map image, ground truth



MANO



Mesh

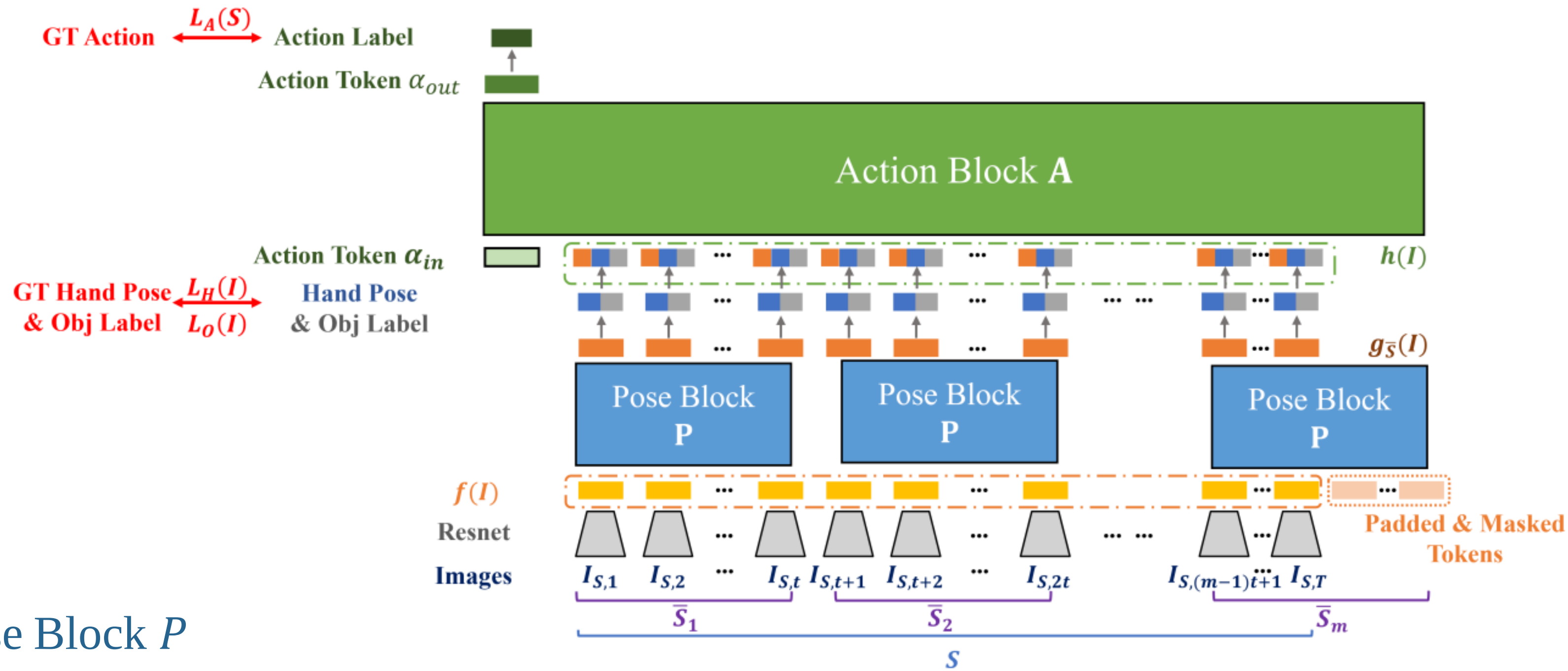
HTT: Hierarchical Temporal Transformer[1)]

- A transformer-based framework to exploit temporal information
  - Challenging task due to self-occlusion and ambiguity to hand motions and actions from egocentric RGB videos.

1) Wen Y, Pan H, Yang L, Pan J, Komura T, Wang W. Hierarchical Temporal Transformer for 3D Hand Pose Estimation and Action Recognition from egocentric RGB Videos. CVPR 2023

HTT[1)]

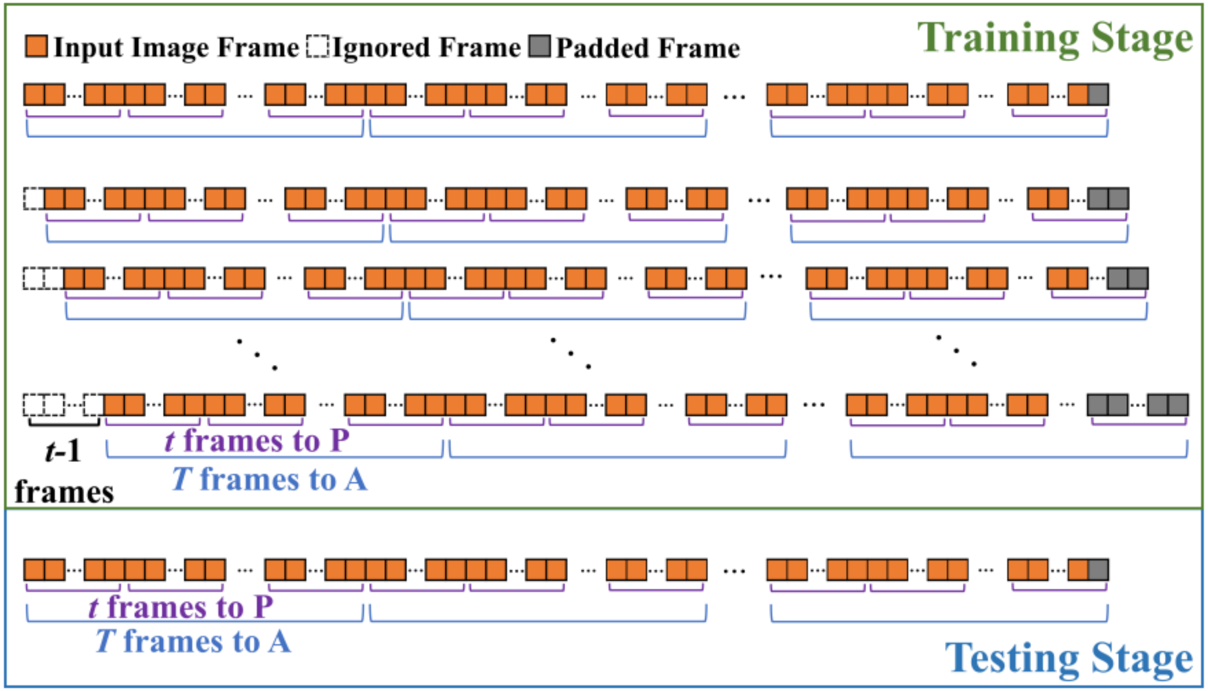
- Overview



  - Pose Block *P*
    - To estimate the pre-framed 3D hand pose and the interacting object category
  - Action Block *A*
    - To aggregate the predicted hand motion and object label over S for action recognition
  - Block Composition
    - 2 transformer encoders, position encoding, attention, normalization, feed-forward

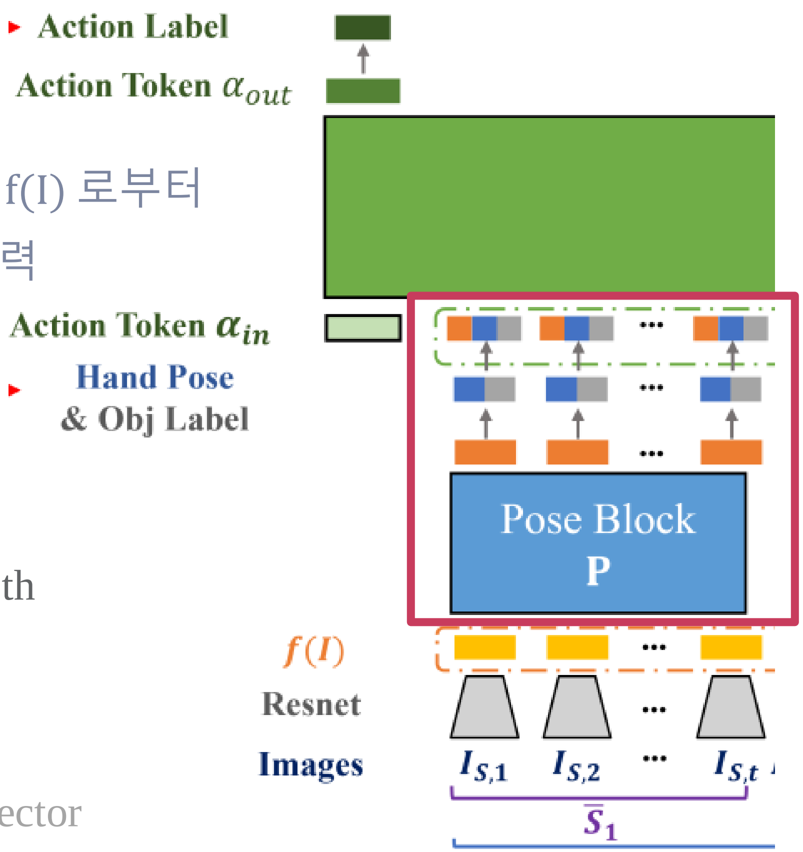
HTT[1)]

- Architecture
  - Pose Block P
    - Shifting window strategy을 사용하여 비디오 클립 S를 m개의 연속적인 segment t개로
      - $seg_t(S) = (\overline{S_1}, \overline{S_2}, \cdots, \overline{S_m})$



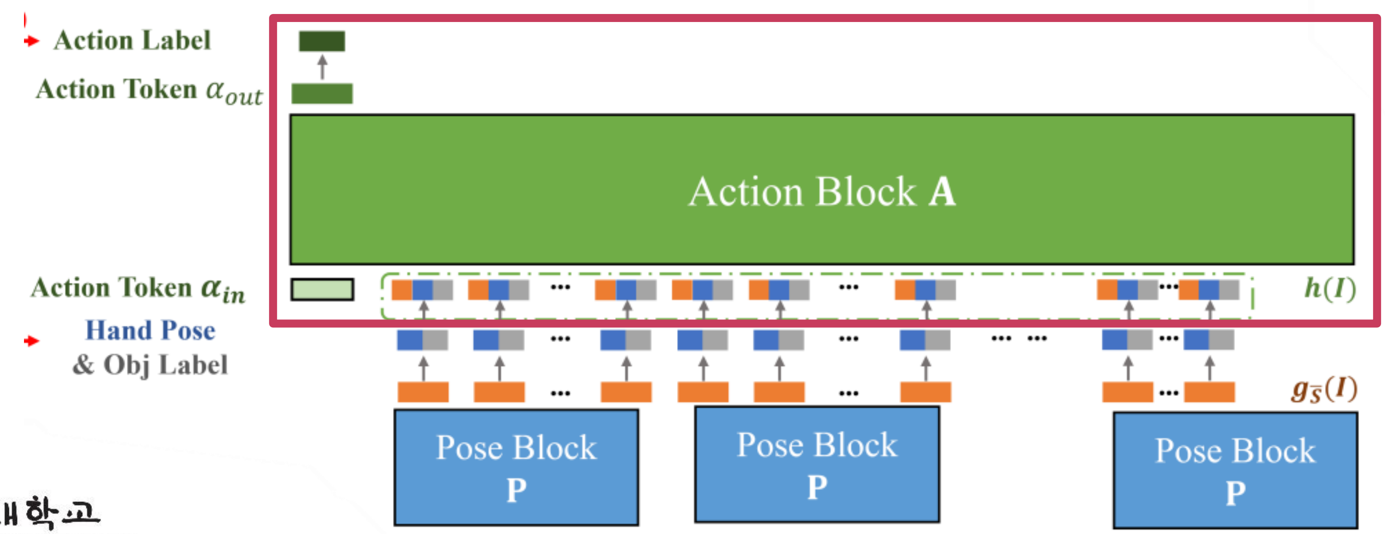
HTT[1)]

- Architecture
  - Pose Block P
    - Temporal cue token $g_{\bar{S}}(I)$ 획득
      - Local segment $\bar{S}$에 해당하는 ResNet features f(I) 로부터 temporal cue를 가진 sequence token $g_{\bar{S}}(I)$ 출력
    - MLP 1을 통해 $g_{\bar{S}}(I)$에서 hand pose $\boldsymbol{P_I}$ 획득
      - $\boldsymbol{P_I} = \left(P_I^{2D}, P_I^{dep}\right) = MLP_1(\boldsymbol{g_{\bar{S}}(I)})$
        - ✓ P_I^{2D}: 2D joint coordinates, P_I^{dep}: joint depth
    - MLP 2를 통해 $g_{\bar{S}}(I)$에서 object label $\mathbf{O_I}$ 획득
      - $\mathbf{O_I}$: n_0 차원 object classification probability vector
      - $\mathbf{O_I} = \left[\mathrm{p}(\mathrm{o}_1|\mathrm{I}), \cdots, p\left(o_{n_o}|I\right)\right] = softmax(MLP_2(\boldsymbol{g_{\bar{S}}(I)}))$



HTT[1)]

- Architecture
  - Action Block A
    - Input sequence S 를 이용하여 action label 예측
      - $\left(\alpha_{in}, h(I_{S,1}), \cdots, h(I_{S,T})\right)$를 Action Block A에 통과시켜 나온 결과 $\boldsymbol{\alpha_{out}}$를 이용하여 probability distribution을 예측
      - $A(S) = [p(a_1|S), \cdots, p(a_{n_a}|S)] = softamx(FC_4(\boldsymbol{\alpha_{out}}))$
        - ✓$\boldsymbol{\alpha_{in}}$: trainable token, action classification을 위한 global information을 aggregation
        - ✓$h(I) = FC_1[FC_2(\boldsymbol{P_1^{2D}}),\ FC_3(\boldsymbol{O_I}),\ \boldsymbol{g_{\bar{S}}(I)}]$



HTT[1)]

- Total training loss
  - $L = L_A(S) + \frac{1}{T}\sum_{S \in seg_t(S)} \sum_{I \in S}(\lambda_2 L_H(I) + \lambda_3 L_O(I))$
    - $L_A(S) = -\sum_{i=1}^{n_a} w_{S,i} \log p(a_i|S)$
      - Cross-entropy loss to classify the action category using target one-hot vector $w(S) = (w_{S,1}, \cdots, w_{S,n_a})$
    - $L_H(I) = \frac{1}{J}(\|P_I^{2D} - P_{I,gt}^{2D}\|_1 + \lambda_1 \|P_I^{dep} - P_{I,gt}^{dep}\|_1)$
      - L1-loss using hyper-parameter λ_1 to balance the different magnitudes of two loss
    - $L_O(I) = -\sum_{i=1}^{n_o} w_{I,i}^o \log p(o_i|I)$
      - Cross-entropy loss for object classification with target probability one-hot vector $w^o(I) = (w_{I,1}^o, \cdots, w_{I,n_a}^o)$
    - λ_2, λ_3: hyperparameters to balance different loss terms

HTT[1)]

- Experiments
  - FPHA, H2O 두 데이터셋 모두 Hand Pose Estimation에서 높은 성능을 보임
    - 3D PCK(Percentage of Correct Keypoints), 3D PCK-RA(root-aligned), MEPE(Mean End-Point Error)

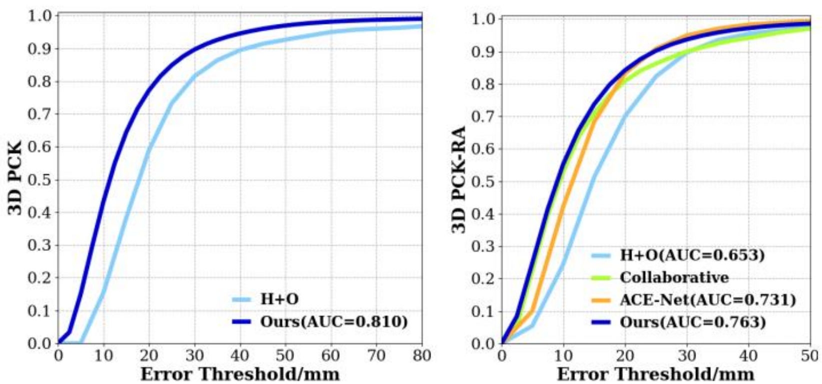


Figure 4. 3D PCK(-RA) of hand pose estimation on FPHA [14]. We report the 3D PCK(-RA) versus different error thresholds by respectively evaluating in the camera space (Left figure) and the root-aligned space (Right figure).

< FPHA 데이터셋의 Hand Pose Estimation 실험 결과 >

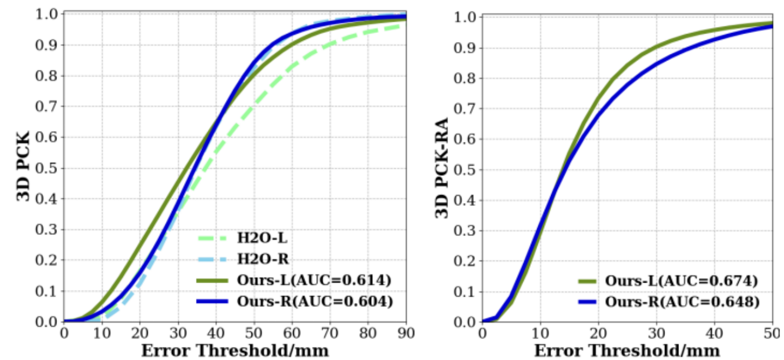


Figure 5. 3D PCK(-RA) of hand pose estimation on the test split of H2O [27]. We report the 3D PCK(-RA) versus different error thresholds by respectively evaluating in the camera space (Left figure) and the root-aligned space (Right figure).

	MEPE in Camera Space				MEPE-RA
	H+O [47]	LPC [18]	H2O [27]	Ours	Ours
Left	41.42	39.56	41.45	**35.02**	16.59
Right	38.86	41.87	37.21	**35.63**	17.91

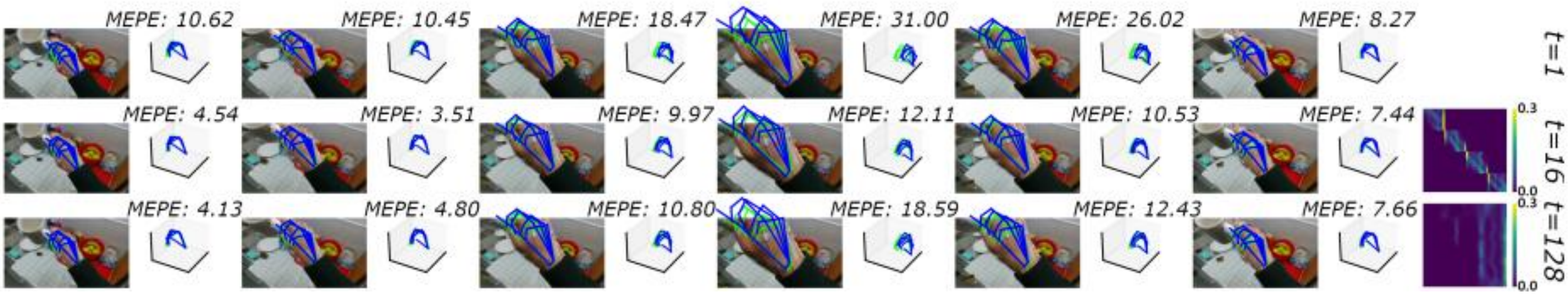
Table 2. MEPE and MEPE-RA of hand pose estimation on the test split of H2O [27], the unit is *mm*.

< H2O 데이터셋의 Hand Pose Estimation 실험 결과 >

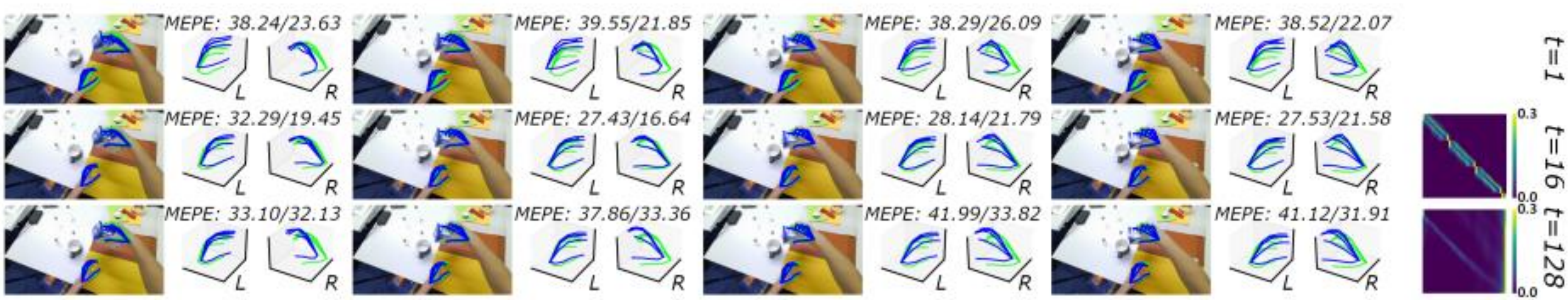
1) Wen Y, Pan H, Yang L, Pan J, Komura T, Wang W. Hierarchical Temporal Transformer for 3D Hand Pose Estimation and Action Recognition from egocentric RGB Videos. CVPR 2023

HTT[1)]

- Ablation Study
  - T=1 & t=16: temporal cue를 이용하면 occlusion이나 truncation에 더 강함
  - T=16 & t=128: long-term temporal cue를 적절하게 이용하면 distant frames에 over-fitting 되는 것을 막고, sharp local motion 보장

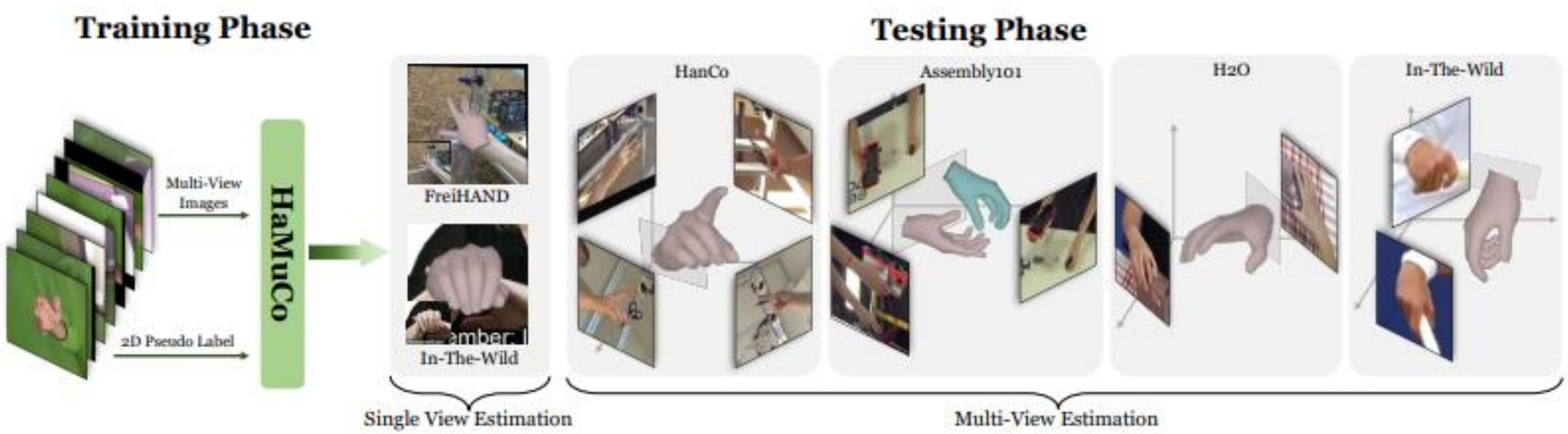


< FPHA 데이터셋의 3D Hand Pose Estimation에 대한 정성적 결과 >



< H2O 데이터셋의 3D Hand Pose Estimation에 대한 정성적 결과 >

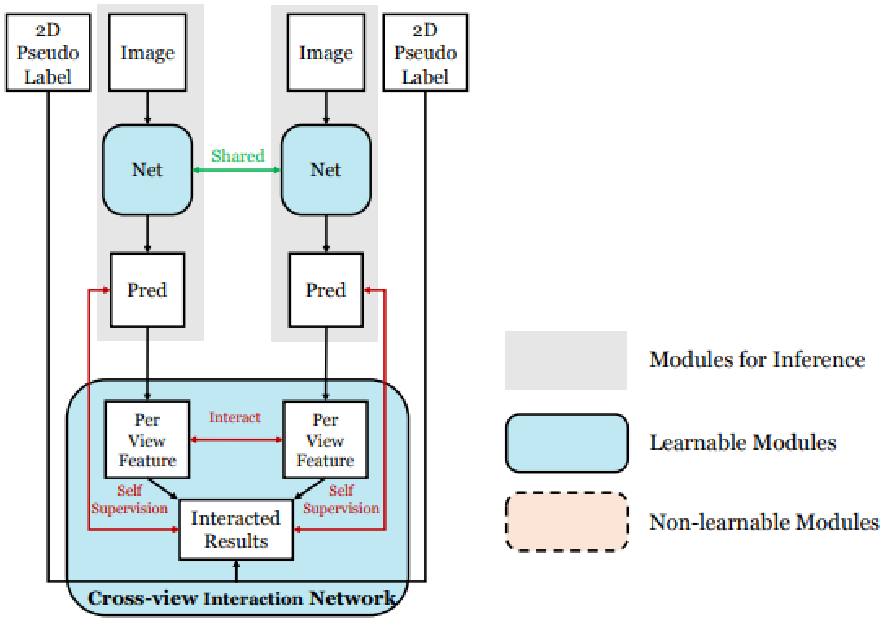
HaMuCo: Hand Pose Estimation via Multiview Collaborative Self-Supervised Learning 2)



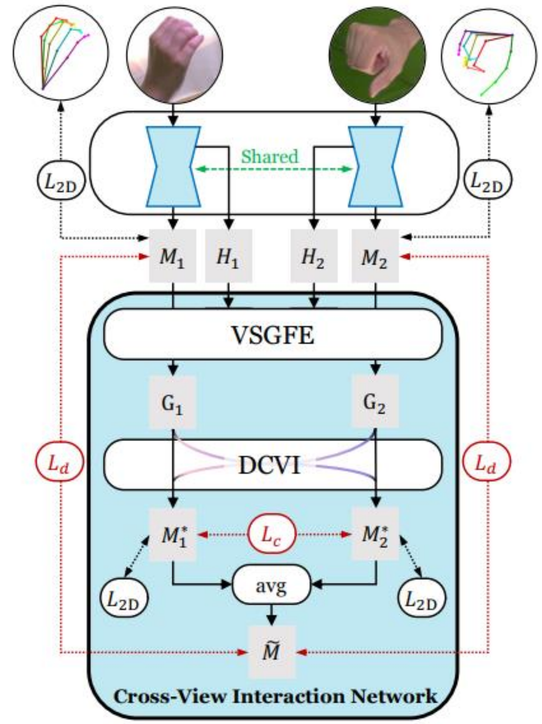
- Self-supervised learning framework that learn single-view hand pose estimator from multi-view pseudo 2D labels
  - Alleviating the label-hungry limitation

HaMuCo[1)]

- Overview



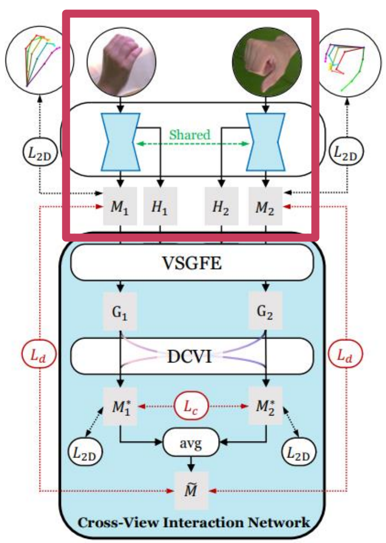
Ours



- Single-View Estimation
  - To extract 3D hand mesh on each view using MANO from multi-view
- Cross-View Interaction Network
  - To capture cross-view features and utilize several consistent losses

HaMuCo[1)]

- Architecture
  - Single-View Estimation
    - Extracting 3D hand mesh $M_i(\theta_i, \beta_i)$ on each view from multi-view synchronized hand images
      - Backbone: for extracting features H^j(after j residual blocks, j=1,2,3,4) from ResNet
      - Regressing Head: for regressing the MANO parameters
      - MANO: for parameters decoding to obtain hand mesh
    - MANO reduces the adverse effects of using poor pseudo labels



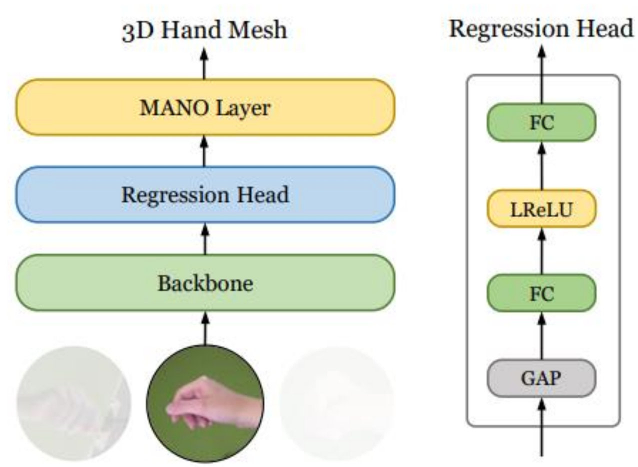
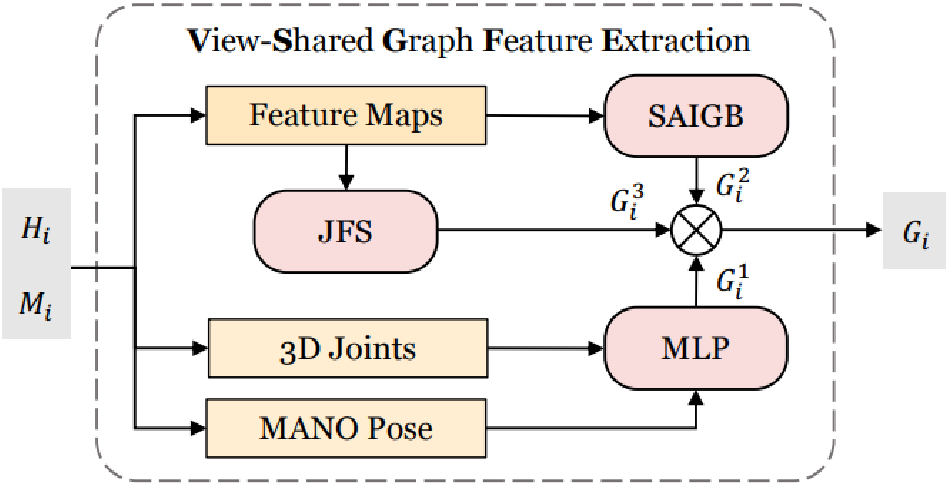


Figure B. The details of our single-view estimation network.

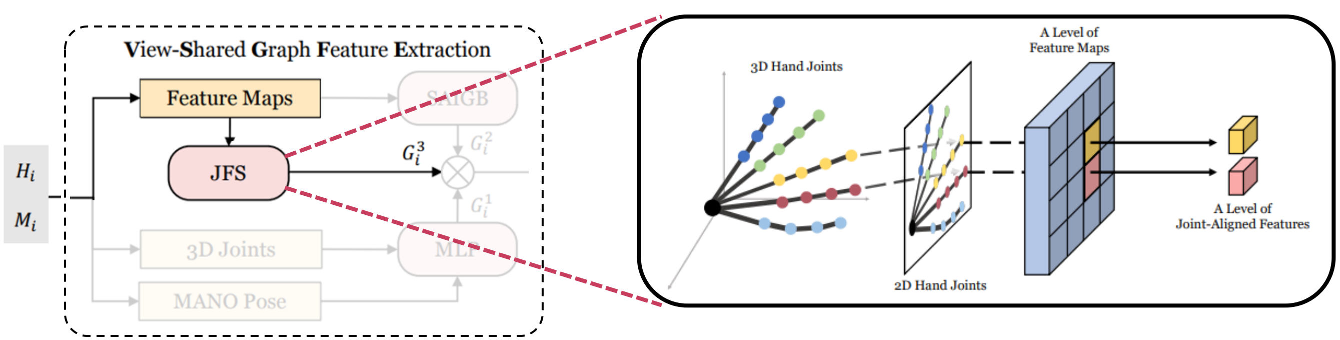
HaMuCo[1)]

- Architecture
  - Cross-View Interaction Network: CVI-Net
    - View-Shared Graph Feature Extraction: **VSGFE**
      - Graph features $G_i = [G_i^1 \otimes G_i^2 \otimes G_i^3]$ 획득
        - ✓Joint embeddings G_i^1
          - explicit geometric information를 포함하는 joint location features 추출
          - location embedding(**LE**) uses MLP to map single-view 3D joints locations P and pose parameter θ



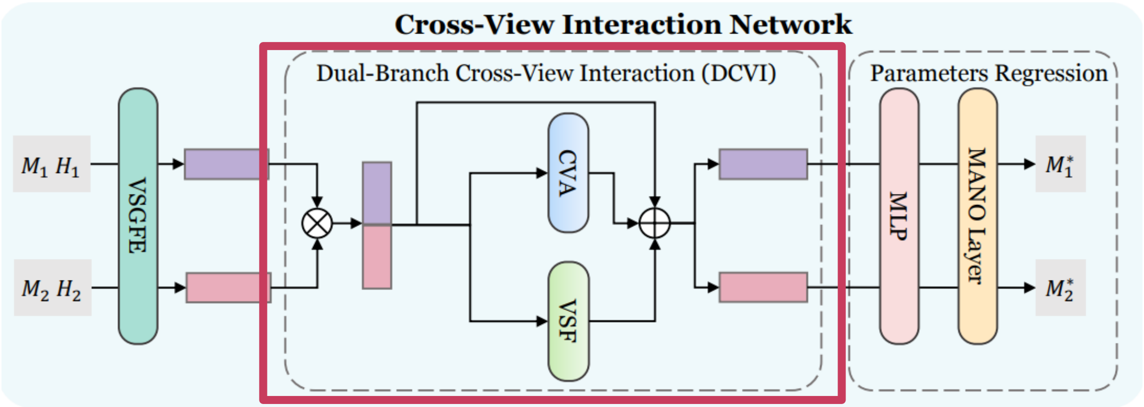
HaMuCo[1)]

- Architecture
  - Cross-View Interaction Network: CVI-Net
    - View-Shared Graph Feature Extraction: **VSGFE**
      - Graph features $G_i = [G_i^1 \otimes G_i^2 \otimes G_i^3]$ 획득
        - ✓Joint-wise high-level image features G_i^2
          - Spatial-Aware Initial Graph Building(**SAIGB**) uses MLP with H^4 and reshape it to get G_i^2
          - Spatial structure information of H^4 을 가진 global image features 추출
        - ✓Joint-aligned features G_i^3
          - Local image features 추출
          - Joint Feature Sampler(**JFS**) projects joints onto multi-level image feature maps $\{H_i^j\}_{j=1}^3$ using camera intrinsics



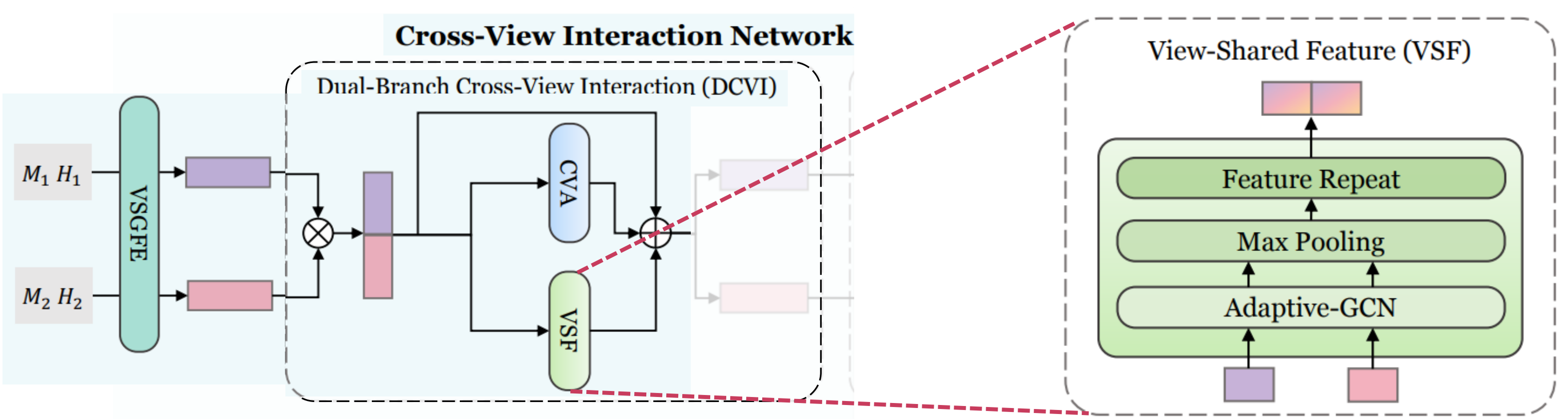
HaMuCo[1)]

- Architecture
  - Cross-View Interaction Network: CVI-Net
    - Dual-Branch Cross-View Interaction: DCVI
      - Complementary information from other views on multi-view graph feature G
      - Cross-View Attention branch: **CVA**
        - ✓multi-view information을 포함하도록 cross-view transformer F_t 이용



HaMuCo[1)]

- Architecture
  - Cross-View Interaction Network
    - Dual-Branch Cross-View Interaction: DCVI
      - View-Shared Feature branch: **VSF**
        - ✓adaptive-GCN F_a 를 이용하여 canonical feature space $C_i = F_a(G_i)$ 획득
          - Node: the hand joints, Edge: joint feature correlation
        - ✓Multi-view $C=\{C_i\}_{i=0}^{v}$에 max-pooling을 적용하여 모든 joint의 max activated features를 포함하는 view-shared features C' 획득
        - ✓View specific feature $G^* = G + F_t(G) + C'$

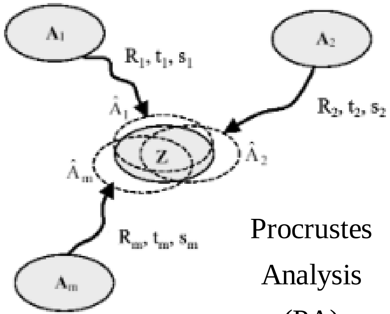


서강대학교 SOGANG UNIVERSITY

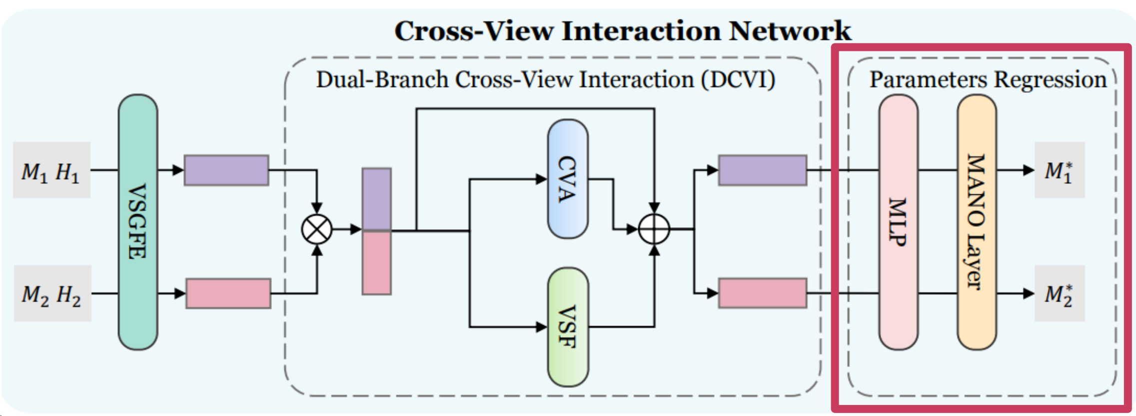
VDS LAB

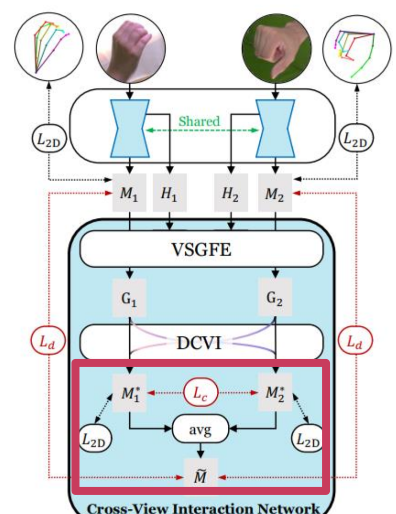
HaMuCo[1)]

- Architecture
  - Cross-View Interaction Network: CVI-Net
    - Parameters regression
      - View specific feature G^*
      - Decoder로 공유된 MLP F_r를 이용하여 pose parameter $\theta^* = F_r(G_i^*)$를 regression
      - Hand mesh $M_i^*(\theta_i^*, \beta_i^*)$, corresponding joints $P_i^* = JM_i^*$
      - $\widetilde{M} = \frac{1}{v}\sum_{i=1}^{v} A(M_i^*)$
        - ✓A: align mesh to a canonical view
          - align with camera pose or Procrustes Analysis



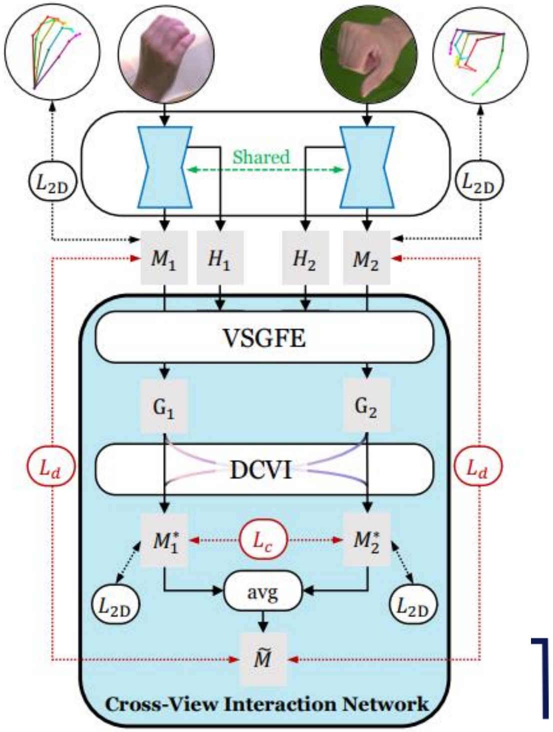
Procrustes Analysis (PA)





HaMuCo[1)]

- Total training loss
  - $L = L_c + L_d + L_{2D} + L_p$
    - $L_c = 2D\ consistency\ loss\ L_{c_{2D}} + Fusion\ consistency\ loss\ L_{c_f}$
      - $L_{c_{2D}} = \frac{1}{v^2}\sum_{i=1}^{v}\sum_{j=1}^{v} \parallel \prod(M_i^*) - \prod(A_i(M_j^*)) \parallel_1$
        - ✓L1 loss to utilize the 2D predictions in every single view to supervise other views
      - $L_{c_f} = \frac{1}{v}\sum_{i=1}^{v} \parallel M_i^* - A_i^{-1}(\widetilde{M}) \parallel_1$
        - ✓L1 loss to use the fused results to supervise each view
    - $L_d = \frac{1}{v}\sum_{i=1}^{v} \parallel M_i - A_i^{-1}(\widetilde{M}) \parallel_1$
      - L1 loss to use the multi-view fusion results to supervise the single-view outputs to achieve self-distillation
    - $L_p = \frac{1}{v}\sum_{i=1}^{v} \alpha(\parallel \theta_i \parallel_1 + \parallel \theta_i^* \parallel_1 + \gamma \parallel \beta_i \parallel_1)$
      - L1 loss to regularize the MANO parameters
        - ✓α, γ : to balance the loss scale
    - L_{2D}: L1 loss to supervise the results from 2D pseudo labels



HaMuCo[1)]

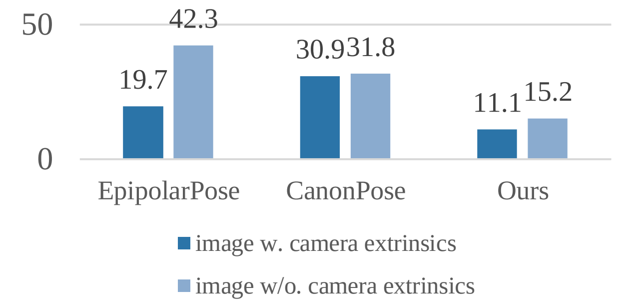
- Experiments on single-view
  - HanCo, FreiHAND 데이터셋 모두 좋은 성능을 보임
    - PA-MPJPE(PA-JE), PA-MPVPE(PA-VE), NMPJPE(N-JE)
    - 훈련 시 camera extrinsics 사용 여부와 관련없이 큰 성과를 보임
    - Backbone 종류에 크게 영향을 받지 않음

Method	Input	N-JE↓	PA-JE↓
Fully-Supervised Method:			
MobRecon [8]	image	9.9	5.7
EpipolarPose [30]	image	10.5	6.1
Self-Supervised Method:			
EpipolarPose [30]	image, 📷	19.7	9.3
CanonPose [56]	2D pose, 📷	30.9	12.6
Ours	image, 📷	**11.1**	**7.0**
EpipolarPose [30]	image	42.3	23.5
CanonPose [56]	2D pose	31.8	12.8
Ours	image	**15.2**	**7.7**

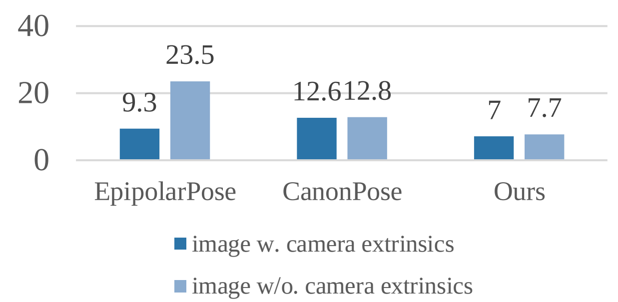
< HanCo 데이터셋의 Hand Pose Estimation 실험 결과 >

Method	Data	Backbone	PA-JE↓	PA-VE↓	F@5↑
Fully-Supervised Method:					
YoutubeHand [31]	Frei.	Res50	8.4	8.6	0.61
I2UV-HandNet [7]	Frei.	Res50	6.7	6.9	0.71
MobRecon [8]	Frei.	Res50†	6.1	6.2	0.76
Ours-SV	Frei.	Res50	7.5	7.5	0.68
Self-Supervised Method:					
S^2HAND [10]	Frei.	EffiNet-b0	11.8	11.9	0.48
Ours-SV	Frei.	EffiNet-b0	11.6	11.7	0.49
Ours-SV	Frei.	Res50	11.9	12.0	0.47
Ours-SV	HanCo	EffiNet-b0	11.3	11.4	0.51
Ours-SV	HanCo	Res50	11.6	11.8	0.48
Ours	HanCo	EffiNet-b0	6.3	6.8	0.71
Ours	HanCo	Res50	**6.2**	**6.7**	**0.72**

< FreiHAND 데이터셋의 Hand Pose Estimation 실험 결과 >



< HanCo 데이터셋의 N-MPJPE 결과 시각화 >



< HanCo 데이터셋의 PA-MPJPE 결과 시각화 >

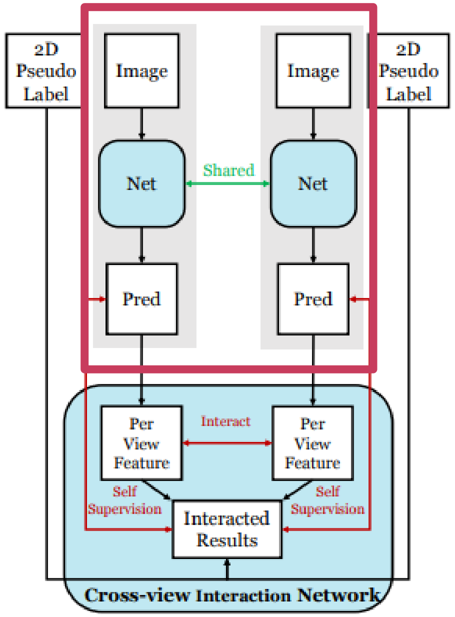
HaMuCo[1)]

- Experiments on multi-view
  - Fully-Supervised method와 비교될 정도로 Self-Supervised method의 성능이 크게 향상됨
    - Opt-Center, RANSAC: for triangulating pseudo labels
    - Opt-Center는 보다 정확한 root-relative results를 제공하기 때문에 PA-MPJPE가 낮음
    - RANSAC는 joint-wise accuracy를 높이기 때문에 MPJPE가 낮음

Method	MPJPE ↓	PA-MPJPE ↓
Traditional Triangulation Method (w/o training):		
DLT [22]	16.8	13.2
Pictorial [12]	13.5	10.2
RANSAC [28]	12.3	9.8
Fully-Supervised Method:		
EpipolarTrans [24]	6.2	4.2
LT-Algebraic [28]	5.5	3.6
LT-Volumetric [28]	5.8	3.6
LT-Volumetric$^+$ [28]	**4.9**	3.6
EpipolarPose$^+$ [30]	8.0	4.4
Ours (Opt-Center)	6.0	**3.2**
Ours (RANSAC)	5.8	3.4

Method	MPJPE ↓	PA-MPJPE ↓
Self-Supervised Method:		
EpipolarTrans [24]	11.2	9.0
LT-Algebraic [28]	10.3	7.8
LT-Volumetric [28]	10.6	8.0
LT-Volumetric$^+$ [28]	9.5	7.2
CanonPose$^+$ [56]	21.6	10.5
EpipolarPose$^+$ [30]	17.2	8.3
Ours (Opt-Center)	8.8	**5.3**
Ours (RANSAC)	**8.5**	5.6

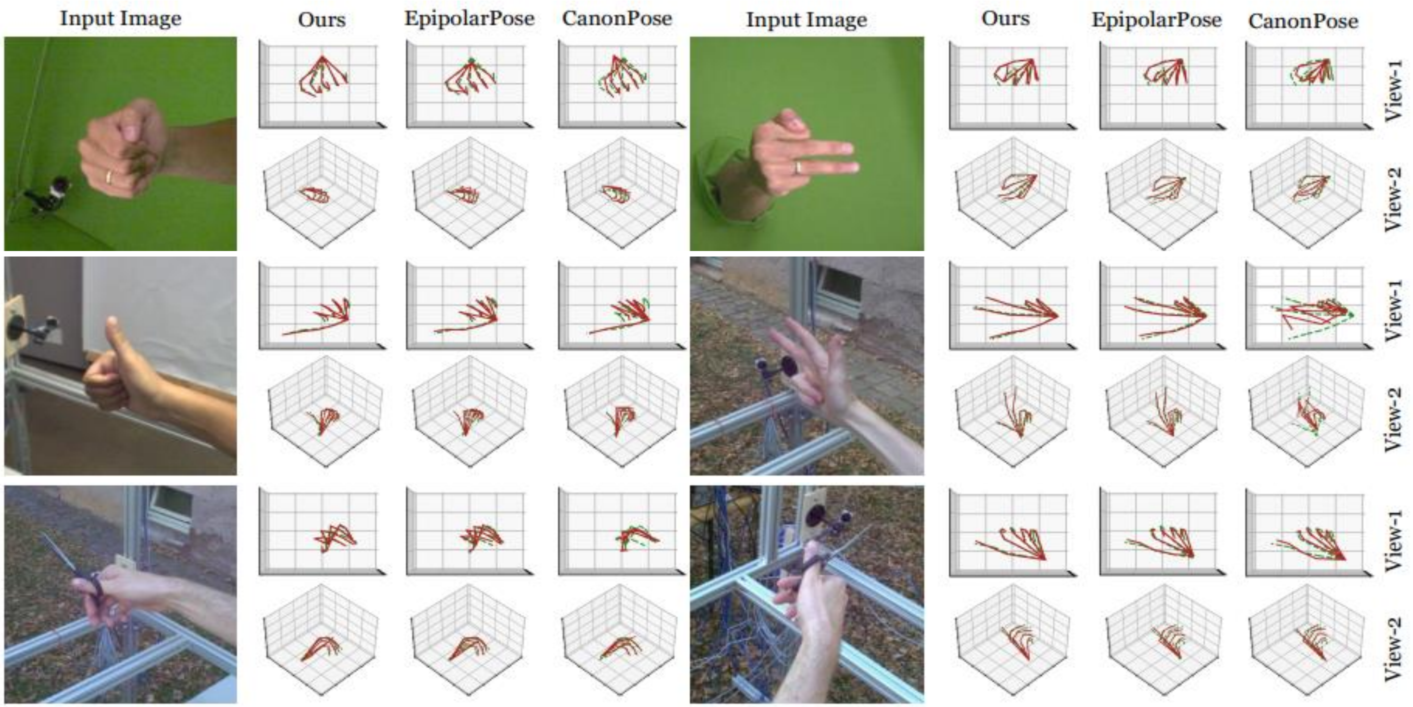
< HaMuCo의 Hand Pose Estimation 실험 결과 >



Ours

HaMuCo[1)]

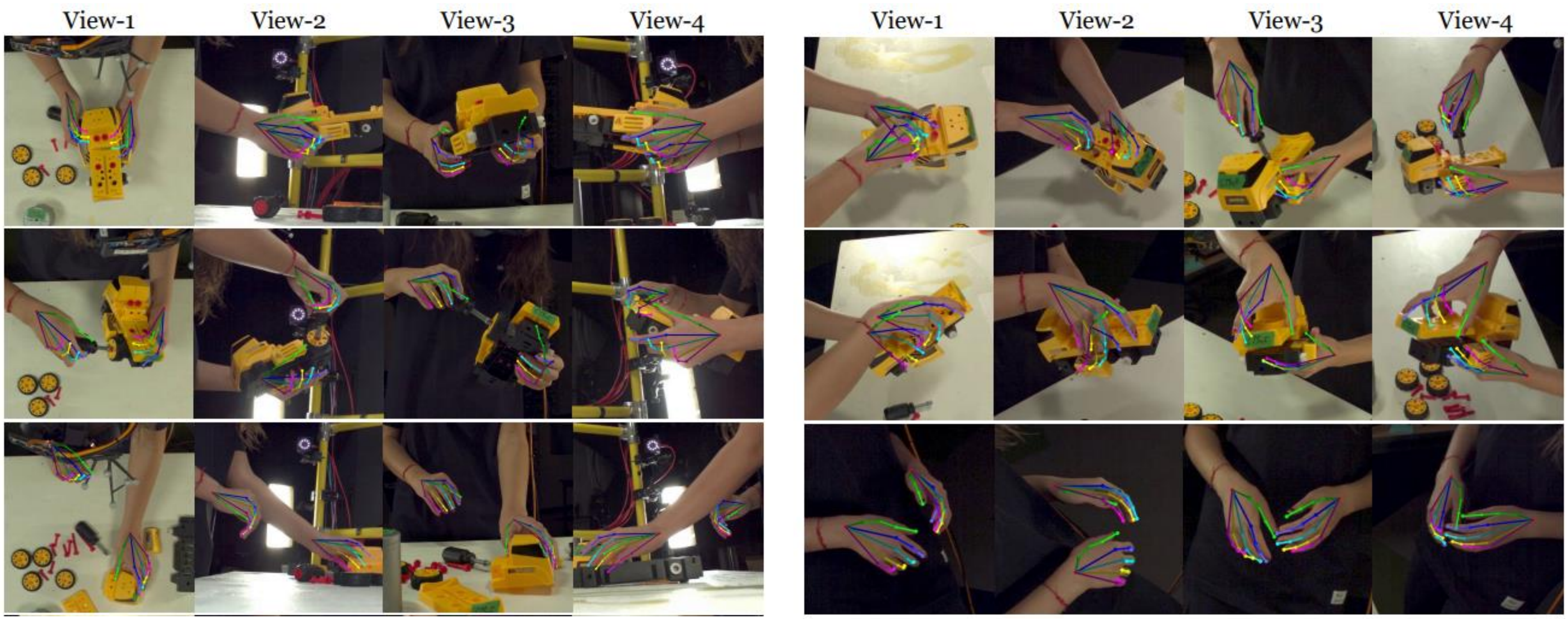
- Experiments on single-view
  - Comparisons between HaMuCo, EpipolarPose, and CanonPose
    - More accurate 3D joints with different gestures, backgrounds, viewpoints, occlusion, object in hands
      - green: ground truth, red: predicted 3D joint keypoints



< HaMuCo, EpipolarPose, CanonPose의 3D Hand Pose Estimation에 대한 정성적 결과 >

HaMuCo[1)]

- Experiments on multi-view
  - Training 시 오른손 데이터만을 사용
    - 손의 수를 알 수 없고, occlusion이 심할 때에도 좋은 성능을 보임



< Assembly101 데이터셋의 3D Hand Pose Estimation에 대한 정성적 결과 >

감사합니다