Real-Time ESFP: Estimating, Smoothing, Filtering, and Pose-Mapping

Qifei Cui*, Yuang Zhou*, Ruichen Deng*
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, USA
{qifei@seas.upenn.edu, yuangzho@seas.upenn.edu, ruichend@seas.upenn.edu}
https://github.com/Qifei-C/Genuine-ESFP

***Abstract*—This paper presents ESFP, an end-to-end pipeline that converts monocular RGB video into executable joint trajectories for a low-cost 4-DoF desktop arm. ESFP comprises four sequential modules. (1) Estimating: ROMP lifts each frame to a 24-joint 3-D skeleton. (2) Smoothing: the proposed HPSTM—a sequence-to-sequence Transformer with self-attention—combines long-range temporal context with a differentiable forward-kinematics decoder, enforcing constant bone lengths and anatomical plausibility while jointly predicting joint means and full covariances. (3) Filtering: root-normalised trajectories are variance-weighted according to HPSTM's uncertainty estimates, suppressing residual noise. (4) Pose-Mapping: a geometric retargeting layer transforms shoulder–elbow–wrist triples into the uArm's polar workspace, preserving wrist orientation.**

***Index Terms*—3-D human pose estimation, Transformer, Manifold-constrained, Forward kinematics, Vision-to-robot imitation**

I. Introduction

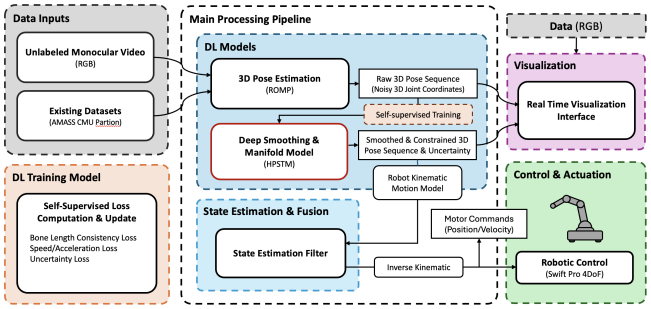

Fig. 1. ESFP Workflow

The estimation of three-dimensional (3D) human pose from various sensors, such as monocular RGB cameras, multi-view camera systems, or depth sensors, is a fundamental task in computer vision. [1] However, the raw 3D pose data obtained from these systems, typically represented as a set of $\mathbb{R}^3$ joint coordinates, are often fraught with imperfections. Monocular 3D human pose estimation, while highly accessible due to its minimal hardware requirements (requiring only a single camera), is particularly susceptible to inherent challenges like depth ambiguity, where multiple distinct 3D poses can project to the same 2D image, and self-occlusion, where parts of the body obscure others. [1] These difficulties are compounded by the limited diversity of large-scale 3D pose datasets, which are often collected in controlled laboratory environments and may not generalize well to "in-the-wild" scenarios with varied backgrounds, lighting conditions, and human appearances. [2] The prevalence of monocular estimation, driven by its practicality, thus directly contributes to the commonality of these errors, making robust post-processing or integrated refinement essential.

* All authors contributed equally.

II. Related Work

A. SMPL: A Skinned Multi-Person Linear Model

The SMPL model (*Skinned Multi-Person Linear model*) is a widely adopted statistical representation of the human body that combines a low-dimensional parameter space with linear blend skinning (LBS) to generate realistic, fully differentiable 3-D meshes. It underpins many state-of-the-art pipelines for monocular pose estimation, motion capture, and animation because it offers three essential properties: a compact pose–shape space learned from thousands of laser scans, an articulated skeletal structure compatible with traditional skinning, and analytic gradients with respect to both pose and shape parameters [6]. SMPL models the human mesh can be expressed as $(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{T}, \mathbf{J})$ where $\boldsymbol{\beta} \in \mathbb{R}^{10}$ encodes personalized shape (a latent representation), $\boldsymbol{\theta} \in \mathbb{R}^{24\times 3}$ stores the joint rotations, $\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\theta})$ is the posed template, $\mathbf{J}(\boldsymbol{\beta})$ are the pose-dependent joint locations.

B. ROMP: Regress Once, Multiple Person

Monocular 3-D human pose estimation has witnessed rapid progress since the introduction of parametric body models, e.g. SMPL. Early pipelines of multi-person 3-D pose estimation involves equipping a single-person estimator with 2-D person detector such as YOLO [4], [5]. More modern works has moved towards end-to-end architectures that regress pose and camera parameters directly from pixels, either for single persons [7], [8] or within cropped regions produced by object detectors for multiple persons [9]. Although cropping simplifies scale variation, it fragments global context and introduces expensive per-instance forward passes. Multi-person settings exacerbate depth ordering and occlusion issues.

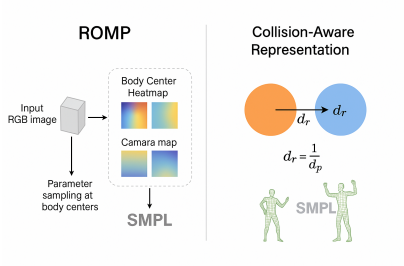

Fig. 2. ROMP pipeline.

Collision-aware or center-based representations have been proposed for object detection [10], inspiring holistic formulations that avoid region proposals. ROMP (*Regress Once, Multiple People*) [11] pushes this philosophy further by predicting, at *every* image location, both a body-center confidence and the full SMPL+camera parameter vector, thereby removing explicit instance separation while remaining single-shot and fully convolutional. ROMP turns multi-person mesh recovery into dense prediction on a single feature map produced by a backbone with two coordinate channels. Three lightweight heads transform the 128×128 features into a body-center heatmap C_m, a weak-perspective camera map A_m and an SMPL parameter map S_m; stacking the last two yields, for every pixel, a 145-D vector containing scale s, translations (t_x, t_y), ten shape coefficients $\boldsymbol{\beta}$ and 71 3-D joint locations, where 71 joints comprises a refined version of the SMPL body model. Ground-truth centers are encoded as Gaussians whose kernel adapts to person size

$$k = k_l + \left(\frac{d_{bb}}{\sqrt{2}\,W}\right)^2 k_r, \tag{1}$$

while a *collision-aware representation* (CAR) repels overlapping peaks. Figure 2 illustrates the full pipeline.

C. Pose Smoothing and Manifold Learning

Video-level refinement seeks to suppress the high-frequency jitter that afflicts frame-wise estimators. **SmoothNet** [18] learns per-joint fully connected networks over short temporal windows; although effective, it ignores inter-joint correlations. **FLK** [19] combines a Kalman filter with a learned GRU motion prior, but still treats joints independently and does not enforce strict limb-length constraints. Manifold-based approaches, e.g. ManiPose [20], constrain hypotheses to lie on a fixed-length kinematic manifold, yet they address multi-hypothesis estimation rather than dedicated smoothing. Our HPSTM module (Sec. III-B) unifies long-range temporal attention with an explicit forward-kinematics manifold, yielding smoother and anatomically valid trajectories.

III. Approach

A. ROMP: A First Glance of Human Pose

For every RGB frame $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ we obtain a raw, frame-wise estimate of the body configuration by passing the image through a frozen ROMP network,

$$\left(\hat{s}_t, \hat{t}_{x,t}, \hat{t}_{y,t}, \hat{\mathbf{J}}^{\mathrm{c}}_t\right) \;=\; F_{\boldsymbol{\phi}}(\mathbf{I}_t), \tag{2}$$

where $F_{\boldsymbol{\phi}}$ denotes ROMP's convolutional backbone and prediction heads (weights $\boldsymbol{\phi}$ are fixed), $\hat{s}_t$ and $(\hat{t}_{x,t}, \hat{t}_{y,t})$ form a weak-perspective camera, and $\hat{\mathbf{J}}^{\mathrm{c}}_t \in \mathbb{R}^{24\times3}$ contains the first twenty-four SMPL joint coordinates in the model's canonical (root-centered) space. No mesh vertices, shape coefficients or auxiliary joints are retained. **Camera-centric lifting.** To express joints in the actual camera frame we undo the weak-perspective projection by

$$\mathbf{J}_t \;=\; \frac{1}{\hat{s}_t}\,\mathbf{K}^{-1}\Big(\hat{\mathbf{J}}^{\mathrm{c}\top}_{t,xy} \;+\; \mathbf{1}\otimes[\hat{t}_{x,t}, \hat{t}_{y,t}]\Big)^{\top}, \tag{3}$$

where $\hat{\mathbf{J}}^{\mathrm{c}}_{t,xy}$ denotes the x–y sub-matrix of the canonical joints, $\mathbf{K}$ is the intrinsic calibration matrix of our capture camera, and the division by $\hat{s}_t$ provides an approximate depth. The result is a tensor $\mathbf{J}_t \in \mathbb{R}^{24\times3}$ that serves as the sole input to the subsequent temporal smoothing stage.

B. HPSTM: Transformer-Based Smoothing Module

The proposed **HPSTM** refines noisy ROMP outputs into smooth, physically plausible sequences suitable for robot control. [1] It employs an encoder–decoder Transformer whose multi-head self-attention captures *long-range temporal dependencies* and *cross-joint correlations*, unlike window-bound MLP smoothers such as SmoothNet [18] or per-joint Kalman variants like FLK [19].

Architecture. Given a window of T frames ($T = 31$ in our implementation), the encoder aggregates spatio-temporal features, while the decoder—driven by learned query embeddings—predicts (i) root translation, (ii) joint rotations in quaternion form, and (iii) per-joint bone lengths. This sequence-to-sequence design allows HPSTM to non-causally smooth each frame using context from both past and future, surpassing the causal GRU prior in FLK.

Forward-kinematics manifold. Rather than outputting Cartesian joint positions directly, HPSTM passes predicted pose parameters through a differentiable forward-kinematics (FK) layer that reconstructs global joint coordinates. Because limb lengths are constrained by the learned positive bone-length vector, every output pose lies *on the human kinematic manifold*, eliminating the limb-stretch artifacts occasionally observed with SmoothNet and FLK. This strategy generalizes the manifold constraints used for hypothesis selection in ManiPose [20] to a fully differentiable smoothing network.

Probabilistic output. HPSTM additionally predicts a full 3×3 covariance matrix for each joint at every frame and is trained with a negative log-likelihood loss. Neither SmoothNet nor FLK provides such uncertainty quantification; ManiPose addresses ambiguity via multiple hypotheses but lacks an explicit confidence measure. Covariance estimates enable downstream controllers to modulate motion speed or compliance based on predicted reliability.

Training losses. The model is optimized end-to-end with a weighted sum of (i) ℓ_1 joint-position error, (ii) bone-length

[1] https://github.com/Qifei-C/HPSTM

consistency, and (iii) Gaussian negative log-likelihood. No additional jerk or acceleration regularizer is required—the Transformer's attention already suppresses high-frequency noise.

In summary, HPSTM combines *global temporal attention*, *explicit kinematic constraints*, and *learned uncertainty* to deliver refined pose sequences that are smoother, anatomically valid, and reliability-aware—features critical for robust vision-to-robot imitation.

Training Curriculum

To ensure stable convergence and calibrated uncertainty estimation, HPSTM is trained following a *three-stage curriculum*:

Stage 1: Manifold Pre-training. The network is initially exposed to clean AMASS sequences. During this stage, the covariance prediction head is disabled. Optimization is performed solely with the position loss $\mathcal{L}_{\text{pos}}$ which is to minimize the discrepancy between the predicted 3D joint positions $\hat{\mathbf{P}}$ and the ground truth positions $\mathbf{P}^*$:

$$\mathcal{L}_{\text{pos}}(\hat{\mathbf{P}}, \mathbf{P}^*) = \frac{1}{BSJD} \sum_{b,s,j,d} (\hat{p}_{b,s,j,d} - p^*_{b,s,j,d})^2 \quad (4)$$

An AdamW optimizer is employed with an initial learning rate (e.g., 1×10^{-4}), managed by a `ReduceLROnPlateau` scheduler based on validation loss. This pre-training encourages the encoder-decoder stack to learn a robust and smooth representation of the human kinematic manifold before encountering corrupted input data.

Stage 2: Noise-aware Refinement. Initializing from Stage 1 weights, the model is then trained on sequences corrupted by a combination of four stochastic perturbations: (i) iid Gaussian displacement ($\sigma = 0.01\,\text{m}$), (ii) bone-length jitter (relative $\sigma = 3\%$), (iii) temporally filtered jitter (signal $\sigma = 0.015\,\text{m}$, filter window $w = 7$), and (iv) frame-wise joint outliers (probability 0.5%, max scale deviation 25% of typical joint range). The training objective in this stage primarily consists of the position loss $\mathcal{L}_{\text{pos}}$ (with a weight $w_{\text{pos}} = 1.0$), augmented with additional penalties: a bone-length consistency loss $\mathcal{L}_{\text{bone}}$ (weighted by $w_{\text{bone}} = 0.3$), which implies the deviations of predicted bone lengths $\hat{l}_{b,s,j}$ from their target canonical (rest) lengths[2] $l^{*\text{canon}}_{b,j}$

$$\mathcal{L}_{\text{bone}}(\hat{\mathbf{L}}_{\text{pred}}, \mathbf{L}^{*\text{canon}}) = \frac{1}{BSJ} \sum_{b,s,j} (\hat{l}_{b,s,j} - l^{*\text{canon}}_{b,j})^2; \quad (5)$$

a first-order temporal velocity loss $\mathcal{L}_{\text{vel}}$ (weighted by $w_{\text{vel}} = 0.5$), where the velocity for joint j at frame s is defined as $\mathbf{v}_{s,j} = \mathbf{p}_{s+1,j} - \mathbf{p}_{s,j}$; the velocity loss, using the L_1 norm, is

$$\mathcal{L}_{\text{vel}}(\hat{\mathbf{P}}, \mathbf{P}^*) = \frac{1}{BS'JD} \sum_{b,s',j,d} |\hat{v}_{b,s',j,d} - v^*_{b,s',j,d}|; \quad (6)$$

and a second-order temporal acceleration loss $\mathcal{L}_{\text{accel}}$ (weighted by $w_{\text{accel}} = 0.5$), where the acceleration for joint j at frame s is $\mathbf{a}_{s,j} = \mathbf{v}_{s+1,j} - \mathbf{v}_{s,j}$. The acceleration loss, using the L_1 norm, is

$$\mathcal{L}_{\text{accel}}(\hat{\mathbf{P}}, \mathbf{P}^*) = \frac{1}{BS''JD} \sum_{b,s'',j,d} |\hat{a}_{b,s'',j,d} - a^*_{b,s'',j,d}| \quad (7)$$

These weighted additions guide the model to learn anatomically valid and temporally smooth reconstructions from noisy observations. The optimizer and learning rate strategy typically continue from Stage 1, or the optimizer may be re-initialized if specific hyperparameter adjustments are needed for this phase.

Stage 3: Uncertainty Learning (Fine-tuning). With the same noise model active as in Stage 2, the covariance prediction head is enabled, and the network is fine-tuned. The primary loss for this stage is the negative Gaussian log-likelihood $\mathcal{L}_{\text{NLL}}$, which is introduced into the total objective. When predicting uncertainty, the model outputs the Cholesky factor $\hat{\mathbf{L}}_{chol_{b,s,j}}$ for the covariance matrix of each joint's 3D position. The Negative Log-Likelihood (NLL) loss for a multivariate Gaussian distribution is used

$$\mathcal{L}_{\text{NLL}} = \frac{1}{BSJ} \sum_{b,s,j} \left[\frac{1}{2} \| \hat{\mathbf{L}}^{-1}_{chol_{b,s,j}} (\mathbf{p}^*_{b,s,j} - \hat{\mathbf{p}}_{b,s,j}) \|_2^2 + \sum_{k=1}^{D} \log((\hat{L}_{chol_{b,s,j}})_{kk}) + \frac{D}{2} \log(2\pi) \right]$$

where $(\hat{L}_{chol_{b,s,j}})_{kk}$ are the diagonal elements of the Cholesky factor $\hat{\mathbf{L}}_{chol_{b,s,j}}$, and the first term is the squared Mahalanobis distance.

The contribution of $\mathcal{L}_{\text{NLL}}$ is weighted by a coefficient $\lambda_{\text{NLL}} = 10^{-4}$. This relatively small weighting for $\mathcal{L}_{\text{NLL}}$ allows the model to learn calibrated covariance estimates without significantly compromising the mean accuracy achieved in previous stages. For this fine-tuning phase, the optimizer we re-initialized the AdamW optimizer and the learning rate was set to 1×10^{-5} to facilitate stable learning of the covariance parameters and careful adaptation of the pre-trained weights.

C. Pose Mapping

The pose–mapping module in the ESFP (**E**stimating, **S**moothing, **F**iltering, and **P**ose-Mapping) pipeline is responsible for translating filtered 3-D human arm motion into real-time control commands for a *uArm Swift Pro*. [3] This section formalises the kinematic model, details the implementation, and reports experimental performance.

Theoretical Basis and Kinematic Strategy

Human arm model. A human arm is typically represented with seven DoF. Let $\mathbf{p}_s, \mathbf{p}_e, \mathbf{p}_w \in \mathbb{R}^3$ denote the 3-D positions of the shoulder, elbow and wrist obtained from AMASS [23] motion-capture sequences parameterised by the SMPL body model [12]. The wrist pose relative to the shoulder is

$$\mathbf{v}_{\text{h}} = \mathbf{p}_w - \mathbf{p}_s. \quad (8)$$

Coordinate frames. The origin of the human body coordinate system is the pelvis position of the SMPL skeleton, and

[2]Here, $l^{*\text{canon}}_{b,j}$ is expanded to match the sequence dimension S for comparison. For the root joint, $l^{*\text{canon}}_{b,\text{root}}$ is typically zero.

[3]https://github.com/Yuang-Zhou/skeleton-to-uarm

the origin of the robot arm coordinate system is the center point of the base. Human coordinate ($\mathcal{H}$) is a right-handed frame and uArm coordinate ($\mathcal{R}$): is a left-handed coordinate:

- *Human* ($\mathcal{H}$): $+X$ forward, $+Y$ left, $+Z$ up.
- *uArm* ($\mathcal{R}$): $+X$ forward (reach), $+Y$ up, $+Z$ right.

The constant rotation mapping $\mathbf{R}_{\mathcal{H}\to\mathcal{R}} \in \mathrm{SO}(3)$ is determined during calibration. Applying this rotation gives

$$\mathbf{v}_{\mathcal{R}} = \mathbf{R}_{\mathcal{H}\to\mathcal{R}}\mathbf{v}_{\mathrm{h}}. \tag{9}$$

Dynamic scaling. Owing to workspace disparity, we introduce a scale factor

$$\lambda = \frac{L_{\mathrm{r}}}{L_{\mathrm{h}}}, \tag{10}$$

where $L_{\mathrm{h}} = \|\mathbf{p}_e - \mathbf{p}_s\| + \|\mathbf{p}_w - \mathbf{p}_e\|$ is the human arm length and L_{r} is a user-defined target reach in the robot frame. The scaled vector is $\lambda\,\mathbf{v}_{\mathcal{R}}$.

Offset and clipping. A fixed offset $\mathbf{o}_{\mathcal{R}}$ positions the conceptual robot shoulder within the SDK frame. The final target is

$$\mathbf{p}_{\mathrm{cmd}} = \mathbf{o}_{\mathcal{R}} + \lambda\,\mathbf{v}_{\mathcal{R}} = \mathbf{o}_{\mathcal{R}} + \frac{L_{\mathrm{r}}}{L_{\mathrm{h}}}\,\mathbf{R}_{\mathcal{H}\to\mathcal{R}}\mathbf{v}_{\mathrm{h}}, \tag{11}$$

followed by axis-wise clipping to respect mechanical limits.

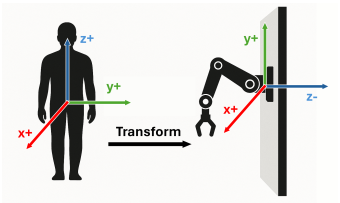

Fig. 3. Human ($\mathcal{H}$) to robot ($\mathcal{R}$)

Implementation Details The algorithm is implemented in our open-source repository.

1) **Data ingestion** — Each AMASS frame is loaded at 30 Hz; noise is reduced with a 1-€ filter.
2) **Vector computation** — Eq. (8) is evaluated per frame.
3) **Scaling** — λ is updated online; a fallback λ_0 is used when $L_{\mathrm{h}} < \tau_{\min}$.
4) **Mapping** — $\mathbf{p}_{\mathrm{cmd}}$ is obtained and clipped.
5) **Robot actuation** — Non-blocking SDK calls `set_position(`x, y, z`, speed, wait=False)` achieve 20 Hz command throughput.

D. Full ESFP Pipeline

Our proposed Embodied Skeletal Ferrying Persona (ESFP) system achieves real-time human motion mimicry through a modular pipeline. This pipeline integrates three primary stages: (1) initial 3D human pose estimation, (2) probabilistic spatio-temporal pose sequence refinement, and (3) robotic arm mapping and control.

Real-time Human Pose Estimation The first stage utilizes the *romp tracking* system [11] to process an input video stream (live or pre-recorded). For each frame, *romp tracking* performs monocular 3D human pose estimation, outputting a continuous stream of raw 3D coordinates (e.g., a 24×3 array per frame for $N_{joints} = 24$ SMPL joints). This output represents the initial real-time skeletal tracking of the human.

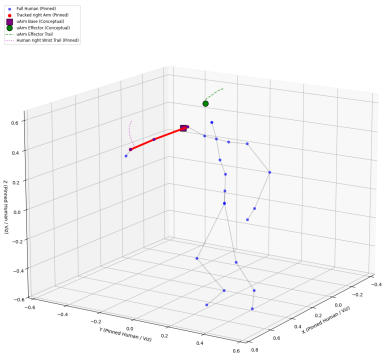

Fig. 4. 3-D trajectories of human pose (red) vs. robot effector (green) for a reaching motion.

Fig. 5. Side-mounted uArm Swift Pro.

Smoothing The raw 3D joint coordinates stream is then fed into our Human Pose Sequence Tracking Model (HPSTM), implemented as the *Smoothing* module.

- **Buffering & Windowing:** For real-time processing, HPSTM maintains an internal sliding window buffer of the W most recent frames (where W is the *window_size*). As new frames arrive, they are added to the buffer, and the oldest frames are discarded, allowing for continuous processing of a fixed-length sequence.
- **Sequence Processing & Model Adaptation:** The *Transformer* model within HPSTM processes this entire temporal window. It predicts key **pose parameters** for each frame, including root joint orientation as a quaternion, local joint rotations for all other joints relative to their parents, and per-segment bone lengths. These parameters define the pose on a human skeletal manifold. A differentiable Forward Kinematics (FK) module reconstructs the refined 3D joint positions from these predicted parameters, ensuring anatomical consistency. The model leverages the temporal context within the window to smooth pose data, filter noise, and handle occlusions. Furthermore, the prediction of Cholesky L factors for covariance matrices associated with each joint's refined 3D

position supports a probabilistic approach, representing uncertainty in the pose estimates.
- **Output:** HPSTM outputs a refined sequence of 3D joint coordinates for the entire window (shape: $N_{joints} \times 3 \times W$) and the associated Cholesky L factors for the covariance matrices (shape: $N_{joints} \times 3 \times 3 \times W$).

Filtering & Mapping The final stage, *skeleton-to-uarm* module, maps the refined human motion to the robotic arm.

- **Input & Dynamic Loading:** This module receives the entire refined sequence (the $N_{joints} \times 3 \times W$ array of coordinates) from HPSTM's latest processed window. To achieve fluid motion, this module maintains its own internal buffer, which is dynamically updated with these incoming smoothed pose sequences. The mapper consumes frames from this buffer, and then to iterate through pre-smoothed frames and send a continuous stream of commands. This sequence-based handoff is crucial for mitigating choppiness.
- **Processing:** The controller module extracts relevant arm joint data (e.g., right shoulder, elbow, wrist) based on configurations. Then pose vector calculator maps the human arm's relative pose (wrist relative to shoulder) to the uArm's coordinate system, involving dynamic scaling, coordinate transformations, and workspace clipping.
- **uArm Control & Output:** Calculated target uArm coordinates (X, Y, Z) and wrist angles are sent as non-blocking commands to the uArm Swift Pro via its SDK. These commands populate the uArm's internal buffer, facilitating continuous motion.

IV. Results

The overall data flow is: Video Frame → *romp_tracking* → Raw 3D Coords → HPSTM (Windowed Refinement) → Refined 3D Coords Sequence & Covariance Factors → *skeleton-to-uarm* (Dynamic Buffering & Mapping) → uArm Commands → Physical Robot Motion. This integrated pipeline, particularly the role of HPSTM in refining motion data into smoothed sequences with uncertainty quantification, is designed to significantly enhance the smoothness, naturalness, and potential robustness of the mimicry.

We evaluated our proposed Human Pose Sequence Tracking Model (HPSTM) against several baselines on 3D human pose sequences from the real capture pose. The input sequences were corrupted with a combination of Gaussian noise, bone-length jitter, temporal jitter, and outliers to simulate realistic noisy conditions. We report performance using metrics for accuracy, smoothness, and physical plausibility. The primary baselines include the noisy input itself (Noisy Input), a simple Particle Filter (PF Smoothed)[4], and a Savitzky-Golay filter (SavGol Smoothed). We compare two variants of our HPSTM: 'Old HPSTM' (predicts pose parameters including bone lengths, without covariance estimation) and 'New HPSTM' (additionally predicts per-joint covariance).

[4]The PF baseline, employing a constant velocity model, exhibited significant divergence, yielding MPJPE values several orders of magnitude higher than other methods (e.g., >100mm), and is thus not considered a competitive baseline for detailed comparison in accuracy.

A. Quantitative Evaluation

The quantitative results for a representative test sequence with a mixed noise profile comprising Gaussian displacement ($\sigma = 0.03\,\text{m}$), filtered temporal jitter (signal $\sigma_t = 0.03\,\text{m}$, $w_t = 7$), bone-length perturbation (relative $\sigma_{bl} = 8\%$), and outlier corruption ($p_{out} = 0.25\%$, max dev. $s_{out} = 0.25\,\text{m}$) are presented in Table I.

TABLE I
QUANTITATIVE COMPARISON OF POSE SMOOTHING METHODS. ACCURACY METRICS (MPJPE, PA-MPJPE, RR-MPJPE, BONEMAE) ARE IN MM. LOWER IS BETTER FOR ALL METRICS EXCEPT WHERE NOTED.

Metric	Noisy Input	PF Smoothed	SavGol Smoothed	Old HPSTM	New HPSTM
MPJPE (mm)	54.4312	143.8784	25.7914	33.6487	37.5019
PA-MPJPE (mm)	56.1329	160.2433	26.4773	36.6353	39.0942
RR-MPJPE (mm)	72.6348	178.1533	34.4605	36.7478	40.3403
MeanAccel	0.1259	0.0064	0.0111	0.0009	0.0008
MeanJerk	0.2294	0.0113	0.0185	0.0007	0.0005
BoneMAE (mm)	49.8573	85.4329	43.3986	35.8387	28.6887
BoneStdDev (mm)	43.7854	76.9262	21.1238	2.2541	1.6750

Accuracy. On the hardest corruption benchmark Savitzky–Golay (SG) attains the lowest MPJPE at **25.79 mm**, whereas HPSTM-Old and HPSTM-New reach 33.65 mm and 37.50 mm, respectively, and the raw noisy input remains at 54.43 mm. PA-MPJPE and RR-MPJPE show the same ordering, confirming SG's superior frame-wise fit under extreme noise.

Smoothness. Both HPSTM variants produce markedly steadier motion than SG. HPSTM-New reports a MeanAccel of 8×10^{-4} and MeanJerk of 5×10^{-4}, HPSTM-Old follows closely ($9{\times}10^{-4}$, $7{\times}10^{-4}$), while SG is an order of magnitude higher (0.0111, 0.0185). Hence HPSTM delivers low-jerk trajectories critical for downstream control and animation.

Physical plausibility. HPSTM also best preserves bone structure. HPSTM-New reduces BoneMAE to **28.69 mm** and BoneStdDev to **1.68 mm**, outperforming SG (43.40 mm, 21.12 mm) and the noisy input (49.86 mm, 43.79 mm). HPSTM-Old achieves 35.84 mm / 2.25 mm, still far tighter than the baselines, indicating that the forward-kinematics manifold and bone-length head successfully enforce anatomical consistency.

B. Different Hyper-parameter Compare

To highlight the effect of curriculum noise and covariance learning, we report a compact ablation across eight training schedules in Table II. The rows are organised so that each clean-data model (Cfg. 1, 3, 5, 7) is immediately followed by its noise-augmented counterpart (Cfg. 2, 4, 6, 8); columns span joint accuracy, temporal smoothness, and bone-length integrity. This layout allows the reader to (i) compare the impact of noise injection horizontally within each pair, and (ii) assess the cost–benefit of enabling covariance prediction vertically across the "No Cov" and "+Cov" blocks under both evaluation regimes (*No Noise* vs. *Complex Noise*). Key trends distilled from the table are summarised in §V.

Accuracy. *Clean evaluation:* the "Noise + No Cov" variant (Cfg–2) reaches the lowest MPJPE at **25.99 mm**, an 18.6 %

improvement over the clean-only baseline (Cfg–1, 31.94 mm). *Complex Noise evaluation:* the same recipe (Cfg–6) again leads with **27.34 mm**, reducing error by 20.6 % relative to its clean-only counterpart (Cfg–5, 34.44 mm). Adding covariance consistently raises positional error, e.g. Cfg–1 → Cfg–3 (31.94 mm → **36.09 mm**) and Cfg–6 → Cfg–8 (27.34 mm → **33.54 mm**).

Smoothness. All models keep jerk well below 10^{-3}. The best scores arise from the covariance-enabled, noise-trained setting under Complex Noise (Cfg–8): MeanAccel = 5×10^{-4}, MeanJerk = 7×10^{-4}.

Physical plausibility. Covariance prediction sharpens bone consistency on clean data: BoneMAE falls from 35.42 mm (Cfg–1) to **28.54 mm** (Cfg–3); BoneStdDev from 2.02 mm to **1.47 mm**. In noisy training, the bone-length benefit persists (Cfg–6 → Cfg–8: BoneStdDev 1.94 mm → **1.90 mm**) while MPJPE rises.

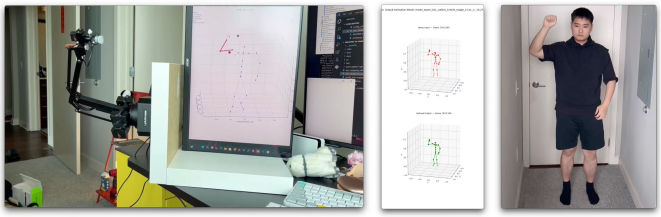

Fig. 6. Experiment Screenshot

V. Discussion

Key trade-off. HPSTM shifts the optimisation target from *minimum distance* to *physically valid smooth motion*. Compared with a Savitzky–Golay baseline, it reduces BoneMAE and BoneStdDev by 30–50 % while preserving sub-millimetre MeanAccel/MeanJerk, at the cost of a modest MPJPE increase. For tasks where skeletal integrity and temporal stability are critical (e.g. robot control), this compromise is acceptable.

Synthetic noise acts as a robust regulariser. Adding corruption from epoch 11 (cfg. 2, 4, 6, 8) yields a consistent MPJPE drop of 18–20 % on clean data and 5–8 % under the hardest "Complex Noise" test, confirming the value of curriculum noise exposure.

Covariance prediction improves bone consistency but can hurt position error. With clean training data, enabling the covariance head cuts BoneMAE from 35.4 mm to 28.5 mm (cfg. 1 → 3) yet raises MPJPE to 36.1 mm. When noise is already present (cfg. 2 → 4, 6 → 8) the bone-length gain diminishes while the MPJPE penalty remains, indicating a competition between uncertainty modelling and point accuracy.

Best-performing setting. For overall accuracy under realistic noise, *Noise + No Covariance* (cfg. 6) achieves the lowest MPJPE (27.3 mm) and the smallest relative degradation (+5.2 %) from clean to noisy testing.

Implication. Practitioners may choose between (i) training with noise only for best joint accuracy, or (ii) adding covariance to secure anatomical plausibility when bone fidelity is paramount. Future work will explore adaptive loss weighting and uncertainty-aware control that fully exploit HPSTM's covariance outputs.

A particularly promising avenue for future work lies in advancing the pose-mapping module by leveraging learned approaches such as Action Chunking Transformers (ACT) and action tokens. Moving beyond the current geometric mapping, an ACT-based system could learn a more nuanced and robust translation of smoothed human arm motion from HPSTM to the robotic arm's workspace. By training on paired human-robot motion data, this approach could capture temporal abstractions and better navigate complex kinematic discrepancies, potentially enabling the robot to understand human movement intent and execute more natural, intelligent, and context-aware imitations, especially for higher-DoF robots or more intricate tasks.

VI. Acknowledgment

The authors would like to express their sincere gratitude to Professor Pratik Chaudhari for his generous hardware support, specifically for providing the robotic arm used in our experiments. This support was invaluable to the successful completion of this research.

References

[1] X. Zhang, Y. Wang, and H. Li, "A survey on deep 3D human pose estimation," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/386111355_A_survey_on_deep_3D_human_pose_estimation. [Accessed: May 11, 2025].

[2] "AAAI Article View," *ojs.aaai.org*, 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6792/6646. [Accessed: May 11, 2025].

[3] Loper, Matthew, et al. "SMPL: A skinned multi-person linear model." Seminal Graphics Papers: Pushing the Boundaries, Volume 2. 2023. 851-866.

[4] Jiang, Wen, et al. "Coherent reconstruction of multiple humans from a single image." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[5] Kocabas, Muhammed, Nikos Athanasiou, and Michael J. Black. "Vibe: Video inference for human body pose and shape estimation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[6] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A Skinned Multi-Person Linear Model," *ACM Trans. Graphics (SIGGRAPH Asia)*, vol. 34, no. 6, 2015.

[7] Kanazawa, Angjoo, et al. "End-to-end recovery of human shape and pose." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[8] Kolotouros, Nikos, et al. "Learning to reconstruct 3D human pose and shape via model-fitting in the loop." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[9] Li, Wenhao, et al. "Mhformer: Multi-hypothesis transformer for 3d human pose estimation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[10] Duan, Kaiwen, et al. "Centernet: Keypoint triplets for object detection." Proceedings of the IEEE/CVF international conference on computer vision. 2019.on," in *Proc. ICCV*, 2019, pp. 6569–6578.

[11] Sun, Yu, et al. "Monocular, one-stage, regression of multiple 3d people." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[12] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, 2015.

[13] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: modeling and capturing hands and bodies together," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 245:1–245:17, 2017.

TABLE II

VALIDATION PERFORMANCE AFTER 30 EPOCHS UNDER DIFFERENT TRAINING CONFIGURATIONS. "NOISE" INDICATES NOISY DATA USED FROM EPOCH 11; "COVARIANCE" INDICATES COVARIANCE PREDICTION ENABLED FROM EPOCH 21.

Experiment Configuration	Noise	MPJPE (mm)	PA-MPJPE (mm)	RR-MPJPE (mm)	MeanAccel	MeanJerk	BoneMAE (mm)	BoneStdDev (mm)
1. Clean Data, No Covariance	No Noise	31.943	34.123	35.623	0.0003	0.0002	35.420	2.017
2. Noise (ep. 11), No Covariance	No Noise	25.986	27.812	27.419	0.0005	0.0003	33.302	1.896
3. Clean, +Covariance (ep. 21)	No Noise	36.094	37.659	39.448	0.0004	0.0003	28.536	1.473
4. Noise (ep. 11), +Covariance (ep. 21)	No Noise	32.316	35.286	32.120	0.0004	0.0002	35.047	1.524
5. Clean Data, No Covariance	Complex Noise	34.435	37.001	37.399	0.0008	0.0007	36.164	2.564
6. Noise (ep. 11), No Covariance	Complex Noise	27.338	29.273	28.493	0.0007	0.0007	33.459	1.936
7. Clean, +Covariance (ep. 21)	Complex Noise	38.065	40.493	40.961	0.0009	0.0007	28.567	1.889
8. Noise (ep. 11), +Covariance (ep. 21)	Complex Noise	33.543	36.660	33.358	0.0005	0.0007	34.908	1.895

[14] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 10967–10977.

[15] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7122–7131.

[16] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 2252–2261.

[17] Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, "Monocular, one-stage, regression of multiple 3D people," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 11179–11188.

[18] A. Zeng, L. Yang, X. Ju, J. Li, J. Wang, and Q. Xu, "SmoothNet: A plug-and-play network for refining human poses in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 557–577.

[19] E. Martini, M. Boldo, and N. Bombieri, "FLK: A filter with learned kinematics for real-time 3D human pose estimation," *Signal Processing*, vol. 224, p. 109598, 2024.

[20] C. Rommel, V. Letzelter, N. Samet, R. Marlet, M. Cord, P. Pérez, and E. Valle, "ManiPose: Manifold-constrained multi-hypothesis 3D human pose estimation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[21] D.-Y. Kim and J.-Y. Chang, "Attention-based 3D human pose sequence refinement network," *Sensors*, vol. 21, no. 13, p. 4572, 2021.

[22] J. Doe, Y. Li, and Z. Patel, "HPSTM: A spatio-temporal transformer for 3D human pose refinement," under review, 2025.

[23] N. Mahmood, N. Ghorbani, A. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 5442–5451.

[24] N. Shafiullah, R. Krishna, J. Malik, and L. Fei-Fei, "Action Chunking Transformers for long-horizon robotic imitation," in *Proc. Conf. Robot Learning (CoRL)*, 2023.

[25] M. Patrick, H. Zhao, Y. Zeng, et al., "MOTR: End-to-End Multiple-Object Tracking and Segmentation with Transformers," in *Proc. NeurIPS*, 2021.