H-MoRe: Learning Human-centric Motion Representation for Action Analysis

Supplementary Material

A. Appendix

A.1. Adaptive Edge Detection from Flow Maps

In Sec. 3.2.2, to refine motion details and incorporate shape information, we build boundary constraint based on human boundaries and edges of flow maps. The human boundaries are detected using a Canny operator. However, the edges of the flow map cannot be extracted using simple operators like the Canny operator. Therefore, we designed an edge detection method with learnable thresholds specifically for flow maps. We define the edges of a flow map $\boldsymbol{M}$ as a series of discrete points, meaning that these points exhibit either intensity or angular discontinuities relative to their neighboring points. Intensity discontinuities indicate significant differences in offset magnitude between a point i and its neighbors j, *e.g.*, the boundary between a moving foreground and a static background. This can be mathematically expressed as:

$$\boldsymbol{s}_I = \left\{ i \in \boldsymbol{M} \mid \left| \|\boldsymbol{M}_i\| - \|\boldsymbol{M}_j\| \right| \geq \vartheta_i \right\}, \tag{10}$$

where ϑ_i represents the learnable intensity threshold. On the other hand, angular discontinuities refer to situations where the angle of the offset between i and j exhibits a significant difference, often occurring between different body patterns. This can be represented as:

$$\boldsymbol{s}_A = \left\{ i \in \boldsymbol{M} \mid \frac{\boldsymbol{M}_i \cdot \boldsymbol{M}_j}{\|\boldsymbol{M}_i\| \cdot \|\boldsymbol{M}_j\|} \geq \vartheta_a \right\}, \tag{11}$$

where ϑ_a represents the learnable angular threshold. Therefore, for any flow map $\boldsymbol{M}$, its edge map $\boldsymbol{s}$ can be formulated as the union of intensity and angular discontinuities:

$$\boldsymbol{s} = \boldsymbol{s}_I \cup \boldsymbol{s}_A. \tag{12}$$

A.2. Patch-Centroid Distance Validation

In Eq. (8), we propose a method to approximate the Chamfer distance using the patch-centroid distance. Here, we provide some validations for this approximation. Based on the following formulation, the Chamfer distance can be approximately transformed, as shown below, under the condition of sufficient curve smoothness, leading to the theoretical conclusion presented in the main text:

$$\begin{aligned}\mathcal{C}\left(\mathcal{P}_{\boldsymbol{s}}, \mathcal{P}_{\boldsymbol{e}}\right) &= \frac{1}{n_p} \sum_{\{\langle i, \hat{j} \rangle\}} \mathcal{D}\left(i, \hat{j}\right) \\ &\approx \mathcal{D}\left[\frac{1}{n_p} \sum_{\{\langle i, \hat{j} \rangle\}} i, \frac{1}{n_p} \sum_{\{\langle i, \hat{j} \rangle\}} \hat{j}\right] \\ &\approx \mathcal{D}\left(\boldsymbol{c}_{\mathcal{P}_s}, \boldsymbol{c}_{\mathcal{P}_e}\right).\end{aligned} \tag{13}$$

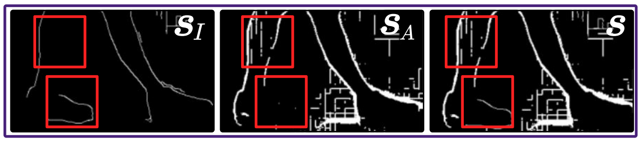

Figure 10. **Edges of flow map.** Through the complementarity of intensity and angular edges (highlighted in the red box), we can effectively detect the edges present in the flow map.

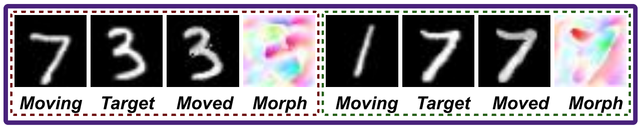

Figure 11. **Validation for patch-centroid distance.** By using the patch-centroid distance as the sole loss function, we can effectively train a network to deform one curve into another by applying an estimated morph.

Apart from the theoretical validation, we also conduct experiments on the MNIST dataset to verify whether the proposed patch-centroid distance can effectively measure the distance between two curves. As shown in Fig. 11, in the experiment, a network tries to apply a non-rigid transformation (Morph) to a moving image (Moving) to generate a moved image (Moved), aligning it with a specified target image (Target). By using the patch-centroid distance as the sole loss function to measure the curve distance between the moved and target images, the network successfully converges. This experimental result further demonstrates the patch-centroid distance as an effective approximation of the Chamfer distance.

A.3. Evaluating Action Recognition with TSN

Similar to Sec. 4.1, besides the results shown in Tab. 2, we also validate the improvement in real-time motion analysis performance achieved by H-MoRe on the action recognition task. Following the same experimental setup as described in Sec. 4.2, we conducted a quantitative comparison on the Diving48 dataset using TSN instead of Video-FocalNets as the action recognition classifier. To validate the improvement in real-time motion analysis performance achieved by H-MoRe, we conducted a quantitative comparison on the Diving48 dataset using TSN as the action recognition classifier. As shown in Tab. 6, compared to using optical flow as the motion representation input, using H-MoRe as the input significantly improved classification performance. This further demonstrates the effectiveness of H-MoRe in real-time scenarios. Besides, due to the use of additional output channels (optical flow: 2 more channels; H-MoRe: 4 more

Methods	Acc@1↑	Acc@5↑	Params (M)
w/o Flow	65.58	95.18	4.5
RAFT	66.09	93.45	4.6 + 5.25
GMA	69.54	94.77	4.6 + 5.88
GMFlow	70.91	95.89	4.6 + 4.68
CRAFT	70.20	95.74	4.6 + 6.30
SKFlow	67.26	94.57	4.6 + 6.27
VideoFlow	71.07	96.80	4.6 + 12.65
FlowFormer++	70.66	95.94	4.6 + 16.15
H-MoRe Ours	**72.69**	**97.60**	4.7 + 5.57

Table 6. **Quantitative comparison for action recognition in real-time scenarios.** Alongside with numbers of learnable parameters in whole recognition pipeline containing motion estimation networks and classifiers.

channels), the number of parameters in classifiers fluctuates slightly compared to the vanilla TSN. However, this does not affect performance. We have also indicated these fluctuations in the tables (Params).

A.4. Skeleton Map Alignment Methods

In Sec. 3.3, we mentioned that to impose a skeleton constraint on the local flow $\boldsymbol{M}_l$, the original skeleton constraint $\mathcal{F}$ needs to be transformed to $\mathcal{F}'$. This is because the $\vec{\boldsymbol{K}}$ used in $\mathcal{F}$ represents the skeleton's offset relative to the environment rather than relative to the subject itself. Therefore, we need to compute the skeleton offset relative to the subject itself by aligning the skeleton map $\boldsymbol{K}_{t+1}$ from frame $\boldsymbol{X}_{t+1}$ to $\boldsymbol{K}_t$ from $\boldsymbol{X}_t$. In practice, we applied two different alignment methods: (i) full-body and (ii) head-anchor. Full-body alignment aligns K_{t+1} by solving the following equation using the least squares method:

$$\begin{aligned} H' &= \arg\min_{H} \left\| H \times \boldsymbol{K}_{t+1} - \boldsymbol{K}_t \right\| , \\ \boldsymbol{K}'_{t+1} &= H' \times \boldsymbol{K}_{t+1}. \end{aligned} \tag{14}$$

Here, H is a homography matrix, which enables skeleton map alignment through projection. This method is suitable for scenarios where the human body does not undergo rotation, such as in gait recognition or video generation.

However, when body rotation occurs, as in diving scenarios, full-body alignment based on all skeletal points may lead to errors in motion estimation. To address this, we use head-anchor alignment. This method employs an affine transform to rotate and scale $\boldsymbol{K}_{t+1}$, ensuring the head regions in the skeleton maps at the two time points are closely matched. Based on this alignment, we obtain the transformed $\boldsymbol{K}'_{t+1}$.

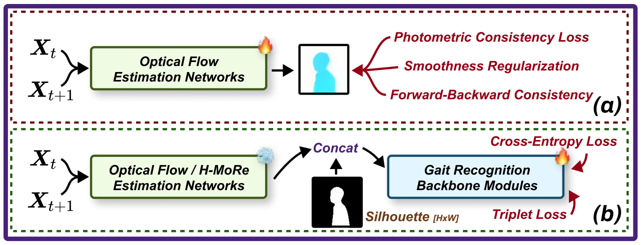
X_t
X_{t+1}
Optical Flow Estimation Networks
Photometric Consistency Loss
Smoothness Regularization
Forward-Backward Consistency
(a)
X_t
X_{t+1}
Optical Flow / H-MoRe Estimation Networks
Concat
Cross-Entropy Loss
Gait Recognition Backbone Modules
Silhouette [HxW]
Triplet Loss
(b)

Figure 12. **Pipeline for gait recognition experiments.** (a) illustrates how we fine-tune the optical flow estimation networks. (b) demonstrates how motion information is integrated into the gait recognition pipeline.

A.5. Details of Experimental Settings

In Sec. 4, we compare the accuracy of H-MoRe in representing motion against optical flows across three tasks. In this subsection, we provide detailed explanations and additional information about our experimental workflow.

A.5.1. Gait Recognition

For the gait recognition task, we first extract $2,800$ sequences from the CASIA-B training set to fine-tune the optical flow estimation models (Fig. 12 (a)) and train our H-MoRe as denoted in Sec. 3.3. After fine-tuning and training, the parameters of these motion estimation networks are frozen. We then use these models as inputs to train classifiers, specifically GaitBase or GaitSet networks with identical structures (Fig. 12 (b)).

Since silhouettes are the default input for gait recognition and require a single-channel input, we adjust the input layer of the classifier for optical flow (2 channels) to support three-channel input. For H-MoRe, which includes world flow (2 channels) and local flow (2 channels), we use a five-channel input. During this stage, only the classifier's parameters are trained. After training, all parameters are fixed, and the models are then tested on the testing set, producing the results shown in Tab. 1.

A.5.2. Action Recognition

For the action recognition task, we similarly extract $14,000$ sequences from the Diving48 training set to fine-tune the optical flow estimation models and train H-MoRe (Fig. 12 (a)). After freezing the parameters of these motion estimation networks, we train downstream classifiers: Video-FocalNets (Tab. 2) and TSN (Tab. 6).

For each moment, RGB images, which are the default input for action recognition, are combined with optical flow or H-MoRe and fed into the classifier. Specifically, these inputs are either (RGB 3 channels + Optical Flow 2 channels) or (RGB 3 channels + H-MoRe 4 channels).

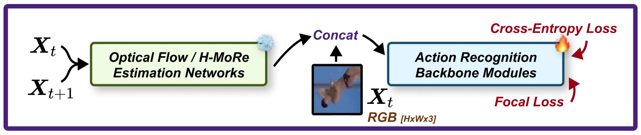
X_t
X_{t+1}
Optical Flow / H-MoRe Estimation Networks
Concat
Action Recognition Backbone Modules
Cross-Entropy Loss
Focal Loss
X_t
RGB [HxWx3]

Figure 13. **Pipeline for action recognition experiments.** Similar to the gait recognition pipeline, the main difference lies in using RGB frames as additional input instead of silhouettes.

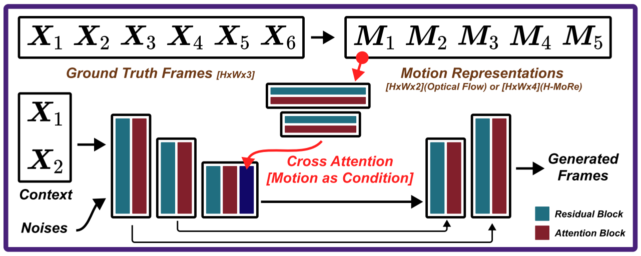
X_1 X_2 X_3 X_4 X_5 X_6
M_1 M_2 M_3 M_4 M_5
Ground Truth Frames [HxWx3]
Motion Representations
[HxWx2](Optical Flow) or [HxWx4](H-MoRe)
X_1
X_2
Context
Noises
Cross Attention
[Motion as Condition]
Generated Frames
Residual Block
Attention Block

Figure 14. **Pipeline for video generation experiments.** This figure illustrates how we use motion representations as conditions for the video generation model to guide video reconstruction. For more details on the network structure, please refer to the LGC-VD.

A.5.3. Video Generation

Our video generation tasks in the main text can be regarded as motion-guided video reconstruction tasks. Unlike the first two tasks, this task is designed to directly evaluate the accuracy of motion provided by H-MoRe. Here, we use the motion at each time step as a condition for a diffusion model to reconstruct the original video. This implies that ignoring lighting variations, the SSIM and FVD scores in Tab. 2 are highly correlated with the accuracy of the motion information provided by optical flow and H-MoRe.

We fine-tune the optical flow estimation models and train H-MoRe using 600 sequences from the UTD-MHAD dataset, then freeze their parameters. Notably, unlike the pipelines in the previous tasks, this task does not concatenate motion directly with input (the first two frames of the sequence). Instead, the motion is used as a condition and integrated into the diffusion models via cross-attention (Fig. 14).

These three tasks collectively demonstrate the effectiveness of H-MoRe in providing accurate motion information, either indirectly (gait recognition and action recognition) or directly (video generation). Additionally, Fig. 7 in the main text and attached video provide a more intuitive visual comparison between our H-MoRe and optical flow, highlighting their respective differences and characteristics.

B. Variables

In the following table, we summarize the symbols used in the main text, along with their detailed definitions and representations.

Variables	Description	Type [shape]
$\boldsymbol{M_w}$	**H-MoRe's world flow.** It represents the offset of each body point relative to the environment, *e.g.*, the ground or the camera.	Matrix $[H \times W \times 2]$
$\boldsymbol{M_l}$	**H-MoRe's local flow.** It represents the offset of each body point relative to the subject themselves.	Matrix $[H \times W \times 2]$
$\boldsymbol{X}_t$ $\boldsymbol{X}_{t+1}$	**Two consecutive frames.** They are usually two frames from a video, spaced more than 0.1 seconds apart.	Matrix $[H \times W \times 3]$
$\boldsymbol{K}_t$ $\boldsymbol{K}_{t+1}$	**Skeleton maps.** They represent a series of (210) skeletal points of a person in frames $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+1}$, with each point containing coordinates and visibility information (c).	Matrix $[210 \times 3]$
$\vec{\boldsymbol{K}}$	**Skeleton offsets.** It represents the offset of each point in the skeleton map between two time steps, *i.e.*, $\boldsymbol{K}_{t+1} - \boldsymbol{K}_t$.	Matrix $[210 \times 3]$
$\boldsymbol{e}$	Human boundaries.	Curve
$\boldsymbol{s}$	Edges of flow map.	Curve
$\mathcal{P}_{\boldsymbol{e}}$	Human boundaries within patch $\mathcal{P}$.	Curve
$\mathcal{P}_{\boldsymbol{s}}$	Edges of flow map within patch $\mathcal{P}$.	Curve
p	Any point within the human body in the flow maps.	Point
$\hat{q}$	The closest point within skeleton offsets $\vec{\boldsymbol{K}}$ with the highest visibility towards p.	Point
i	Any point on the edges of flow map $\boldsymbol{s}$.	Point
$\hat{j}$	The nearest point within human boundaries $\boldsymbol{e}$ towards i.	Point
$\boldsymbol{c}_{\mathcal{P}_{\boldsymbol{e}}}$	The centroid of curve $\mathcal{P}_{\boldsymbol{e}}$.	Point
$\boldsymbol{c}_{\mathcal{P}_{\boldsymbol{s}}}$	The centroid of curve $\mathcal{P}_{\boldsymbol{s}}$.	Point
$\boldsymbol{v_s}$	**Subject motion.** It represents the overall motion trend of subjects.	Vector $[\Delta x, \Delta y]$
$\boldsymbol{u_p}$	The estimated flow $\boldsymbol{M}$ at point p.	Vector $[\Delta x, \Delta y]$
$\boldsymbol{k_{\hat{q}}}$	The skeleton offset $\vec{\boldsymbol{K}}$ at point $\hat{q}$.	Vector $[\Delta x, \Delta y]$
Φ	**H-MoRe's network component.** It estimates world flow $\boldsymbol{M_w}$ between consecutive frames $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+1}$.	Network [Params: $\approx 3.4M$]
Ψ	**H-MoRe's network component.** It estimates subject motion $\boldsymbol{v_s}$ based on world flow $\boldsymbol{M_w}$.	Network [Params: $\approx 2.1M$]
$\mathcal{F}$	**Our skeleton constraint.** It ensures that each body point's movement adheres to kinematic constraints.	Function
$\mathcal{G}$	**Our boundary constraint.** It aligns human shapes onto our estimated flow maps.	Function
$\mathcal{F}_{\mathcal{A}}$	**Angular constraint.** Component of skeleton constraint.	Function
$\mathcal{F}_{\mathcal{I}}$	**Intensity constraint.** Component of skeleton constraint.	Function
$\mathcal{C}$	Chamfer distance between two curves.	Function
$\mathcal{D}$	Euclidean distance between two points.	Function
ω_1	Learnable parameters in Φ.	Parameters
ω_2	Learnable parameters in Ψ.	Parameters
ϑ_a	Threshold for angular constraint $\mathcal{F}_A$.	Constant
ϑ_i^l ϑ_i^h	Low and high boundary threshold for intensity constraint $\mathcal{F}_I$.	Constant

Table 7. **Symbols used in the main text.** Additionally, the description includes its relevant information.