Supplemental Document of Paper "COIN: Control-Inpainting Diffusion Prior for Human and Camera Motion Estimation"

Jiefeng Li[1,2], Ye Yuan[1], Davis Rempe[1], Haotian Zhang[1], Pavlo Molchanov[1], Cewu Lu[2], Jan Kautz[1], and Umar Iqbal[1]

[1]NVIDIA [2]Shanghai Jiao Tong University

In the supplemental document, we provide:

§ A Detailed architecture and training settings of the controlled denoiser.
§ B Ablation study on the EMDB [2] and RICH [1] datasets.
§ C Details of global optimization.

A Controlled Denoiser

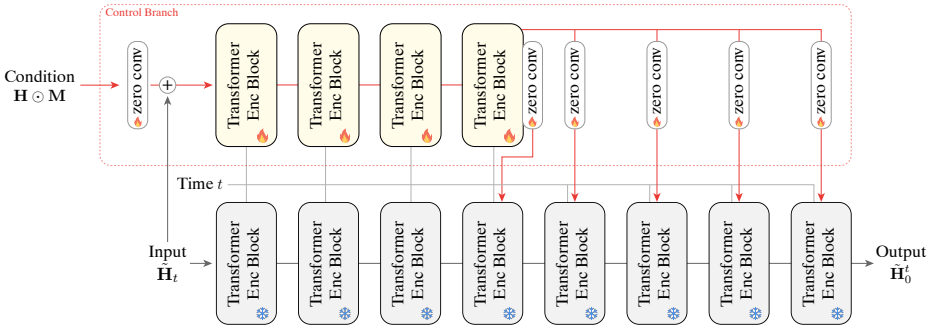

Fig. 1: Architecture of the controlled denoiser.

Architecture Details. The detailed architecture of the proposed controlled denoiser is illustrated in Fig. 1. We adopt a pre-trained transformer-based motion diffusion model as our backbone model. We create a trainable copy of the first 4 encoder blocks. The condition is first encoded by a zero-initialized convolution layer and then concatenates with the input latent motion $\tilde{\mathbf{H}}_t$. The outputs are followed by 5 zero convolution layers and added to the last 5 encoder blocks.

Training Details. We use the AMASS [4] dataset to train the controlled denoiser. To simulate noisy motions in our application, we add Gaussian noise to the conditions. For the root orientation, the noise level is set to 0.05. For the

body pose, the noise level is set to 0.01. For the translation, the noise level is set to 0.1. To simulate occlusions, we randomly mask the conditions. With 0.5 probability, all global trajectories are masked out; with 0.5 probability, all global root orientations are masked out. The probabilities of the above two cases are calculated independently, *i.e.*, it is possible to mask out the trajectories and orientations at the same time. With 0.2 probability, the lower half of the body is masked out; with 0.2 probability, the entire local pose is masked out; with 0.5 probability, we randomly mask body joints, and each joint is masked with 0.3 probability.

B Ablation Study

Impact of Each Component. To comprehensively evaluate the impact of each component of COIN, we further conduct ablation studies on the EMDB [2] and HCM [3] datasets. Quantitative results are shown in Tabs. 1 and 3. COIN shows consistent improvement against other diffusion-based baselines.

Table 1: Global human motion estimation on the EMDB dataset.

Method	PA-MPJPE ↓	W-MPJPE$_{100}$ ↓	WA-MPJPE$_{100}$ ↓	RTE ↓	ROE ↓
Noise Optimization	53.9	873.8	275.8	10.4	96.4
Guided Sampling	107.5	1713.9	462.8	7.2	71.5
Vanilla SDS	64.5	1310.3	520.0	12.3	83.0
COIN w/o Controlled Sampling	39.6	815.2	338.7	7.8	44.3
COIN w/o Dynamic Control	36.4	441.2	162.1	4.1	40.2
COIN w/o Soft Inpainting	35.1	495.4	176.8	4.8	43.6
COIN w/o $\mathcal{L}_{\text{HSR}}$	33.0	461.3	162.6	4.0	38.4
COIN	**32.7**	**407.3**	**152.8**	**3.5**	**34.1**

To further evaluate the effect of each individual loss, we report the W-MPJPE on the RICH dataset.

COIN (full)	w/o $\mathcal{L}_{2D}$	w/o $\mathcal{L}_{3D}$	w/o $\mathcal{L}_{\beta}$	w/o $\mathcal{L}_{smooth}$	w/o $\mathcal{L}_{contact}$	w/o $\mathcal{L}_{SDS}$	w/o $\mathcal{L}_{HSR}$
254.5	448.7	329.4	279.9	256.1	270.3	480.6	273.0

Error Distribution. We further present the error distribution on the EMDB dataset to show more details of the COIN predictions. We also plot the error distribution of WHAM [5] for comparison. The scatter plot is shown in Fig. 2. Here we follow the evaluation protocol of EMDB and evaluate W-MPJPE and WA-MPJPE per 100 frames. It is shown that COIN is more robust and has fewer outlier predictions.

SLAM vs SfM. Compared to SLAM, SfM methods are stronger baselines for camera motion estimation. Because our cases always contain dynamic objects, we

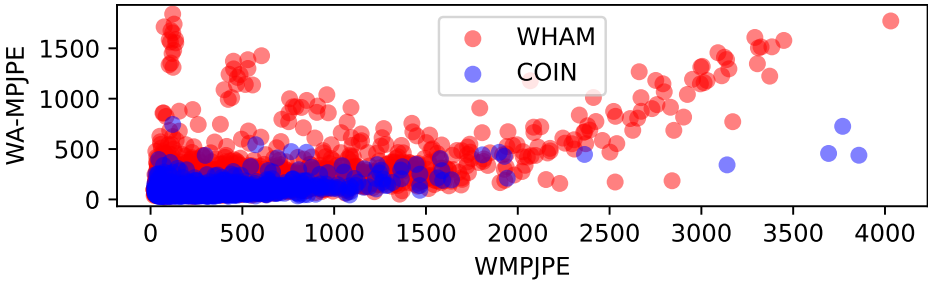

Fig. 2: Error distributions on the EMDB dataset.

used ParticleSfM [6] for its ability to handle dynamic objects and compared it to SLAM. Comparisons are shown in Tab. 2. Note that SfM methods run extremely slow compared to SLAM. Given a video with 700 frames on the RICH dataset [1], ParticleSfM takes over 17 hours while DROID-SLAM only needs 4 minutes. For videos on the EMDB dataset with more than 2000 frames, ParticleSfM took a few days to finish, and could not converge for many. Hence, SLAM is a more viable choice for in-the-wild videos. If we replace DROID-SLAM with ParticleSfM, on the converged videos, the baseline results improve by 200 mm. However, it has minimal impact on COIN demonstrating the robustness of our method to SLAM errors. We would like the emphasize that ParticleSfM could not converge on many of the EMDB videos and the results below are only the subset where it converged.

Table 2: SLAM vs. ParticleSfM on the converged subset of the EMDB dataset.

	HybrIK + SLAM	HybrIK + SfM	COIN (SLAM)	COIN (SfM)
W-MPJPE	643.9	439.0	350.1	330.9

C Global Optimization

Here we detail our optimization formulation for the reconstruction of global human and camera motion. Simultaneously optimizing both camera motion and global human motion can result in local minima. To address this challenge, we follow PACE [3] and adopt a multi-stage optimization pipeline. Before running optimization, we initialize the global human motion with the noisy observations using the controlled denoiser. We randomly sample a Gaussian noise and run DDPM to generate the global motion. In stage 1, we optimize only the first frame camera parameters (R_0, h_0), camera scale s, and the body shape β. In stage 2, we optimize the first frame camera parameters (R_0, h_0), camera scale

Table 3: Global human motion estimation on the HCM dataset.

Method	PA-MPJPE ↓	W-MPJPE ↓	WA-MPJPE ↓	W-RJE ↓	ACCEL ↓
Noise Optimization	66.0	813.9	328.9	794.7	10.2
Guided Sampling	118.2	1653.0	635.4	1626.7	24.7
Vanilla SDS	59.0	1108.2	569.2	1102.6	11.8
COIN w/o Controlled Sampling	47.6	904.8	428.5	898.9	11.0
COIN w/o Dynamic Control	47.4	486.9	239.7	477.5	10.2
COIN w/o Soft Inpainting	48.6	487.1	264.1	478.0	10.8
COIN w/o $\mathcal{L}_{\text{HSR}}$	47.0	488.5	219.3	479.4	10.1
COIN	**45.5**	**479.9**	**212.1**	**470.7**	**10.1**

s, the body shape β, and the global human motion $\mathbf{H}$. In stage 3, we jointly optimize the full camera trajectory along with the global human motion. Given a long video, we split it into windows of $T = 128$ frames. We use 16 overlapping frames to help reduce discontinuities across windows. The mask $\mathbf{M}$ is defined by thresholding the confidence scores of the detected 2D keypoints. The threshold is 0.3. Each stage is run for 500 steps. The learning rates of the 3 stages are 0.01, 0.01, and 0.001, respectively. We use the Adam solver for optimization. Implementation is in PyTorch.

References

1. Huang, C.H.P., Yi, H., Höschle, M., Safroshkin, M., Alexiadis, T., Polikovsky, S., Scharstein, D., Black, M.J.: Capturing and inferring dense full-body human-scene contact. In: CVPR (2022) 1, 3
2. Kaufmann, M., Song, J., Guo, C., Shen, K., Jiang, T., Tang, C., Zárate, J.J., Hilliges, O.: EMDB: The Electromagnetic Database of Global 3D Human Pose and Shape in the Wild. In: ICCV (2023) 1, 2
3. Kocabas, M., Yuan, Y., Molchanov, P., Guo, Y., Black, M.J., Hilliges, O., Kautz, J., Iqbal, U.: PACE: Human and motion estimation from in-the-wild videos. In: 3DV (2024) 2, 3
4. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: ICCV (2019) 1
5. Shin, S., Kim, J., Halilaj, E., Black, M.J.: Wham: Reconstructing world-grounded humans with accurate 3d motion. In: CVPR (2024) 2
6. Zhao, W., Liu, S., Guo, H., Wang, W., Liu, Y.J.: Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In: European conference on computer vision (ECCV) (2022) 3