Shoe Style-Invariant and Ground-Aware Learning for Dense Foot Contact Estimation

Supplementary Material

In this supplementary material, we provide additional technical details and experimental results that were omitted from the main manuscript due to space constraints. The contents are summarized below:

S1. Configuration of COFE dataset

We construct the COFE dataset by aggregating foot image samples from OpenPose [3], InstaVariety [15], PennAction [29], and MPII [1]. Table S1 presents the train–test split of the aggregated dataset, and Figure S1 illustrates the relative proportion of training samples from each source. Regarding the training set, COFE dataset contains 43.3% samples from OpenPose, 38.7% from PennAction, 11.9% from InstaVariety, and 6.1% from MPII.

For image-based datasets of OpenPose and MPII, we only include frames in which the feet are clearly visible. For video-based datasets of PennAction and InstaVariety, we additionally filter out static clips, particularly those of which the person remains standing upright throughout the sequence, as fully contacted foot states are already sufficiently covered in image datasets. To ensure annotation quality, all samples are manually labeled following the OpenPose foot keypoint definition (big toe, small toe, and heel). Moreover, we primarily use videos containing a single person, so as to avoid erroneous keypoint detections caused by multiple individuals within a frame. Notably, the MPII videos are generally very short (less than three seconds), which limits their utility as test set of COFE dataset. Hence, we primarily use MPII dataset only for training.

Although OpenPose includes a small set of test samples, we found that nearly all of them correspond to fully contacted feet. As a result, the evaluation is biased: for example, when predicting uniformly full contact across the entire OpenPose test set, the resulting precision, recall, and F1-score are already 0.712, 0.858, and 0.760, respectively. This suggests that OpenPose is not suitable for robust evaluation. Also, since previous methods for foot contact estimation are video-based and temporal in nature, we choose to perform evaluation solely on the InstaVariety test split samples within the COFE dataset. This choice allows us to coherently evaluate joint-level foot contact estimation.

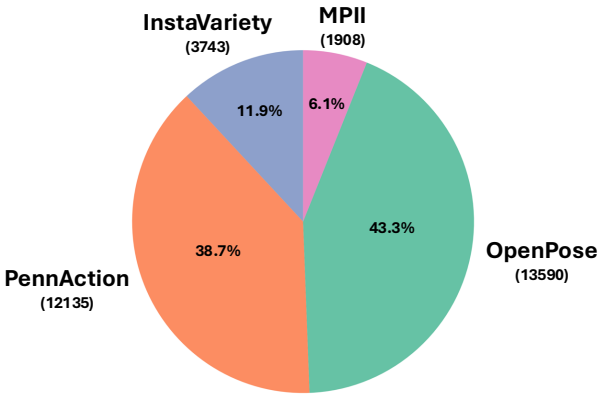

Figure S1. **COFE dataset statistics.** We visualize the dataset configuration of our proposed COFE dataset, which consists of foot image samples in OpenPose [3], InstaVariety [15], PennAction [29], and MPII [1]. We only include training samples.

Table S1. **Data split for COFE dataset.**

Dataset	Image / Video	Train	Test
OpenPose [3]	Image	13,590	464
PennAction [29]	Video	12,135	-
InstaVariety [15]	Video	3,743	1,103
MPII [1]	Image	1,908	-

As shown in Figure S2, the COFE training set maintains a relatively balanced distribution between contacting and non-contacting joints, with 49,927 contacting and 44,201 non-contacting cases overall. At the joint level, big toe contacts account for 63.4% of its annotations, small toe for 53.3%, while heel contacts are slightly underrepresented at 42.5%. This indicates that, the heel joint is slightly less frequent in contact compared to the toes, but the dataset still provides sufficient coverage across all three joints.

When compared with the overall distribution from other training datasets used for FECO (excluding COFE), a clearer contrast emerges. As visualized in Figure S3, the combined datasets exhibit a skew toward non-contact, with roughly 41% contacting versus 59% non-contacting joints in total. Moreover, these datasets show contacts concentrated at the big and small toes, with the heel being signifi-

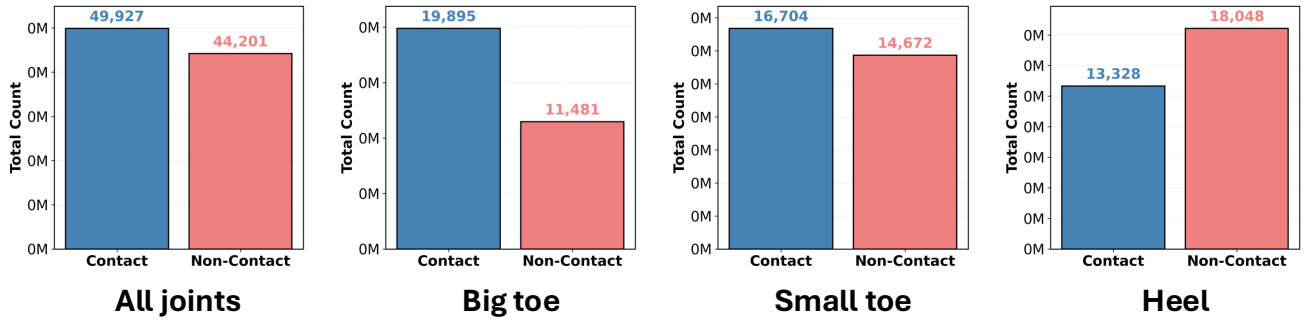

Figure S2. **Contact and non-contact distribution of COFE dataset.**

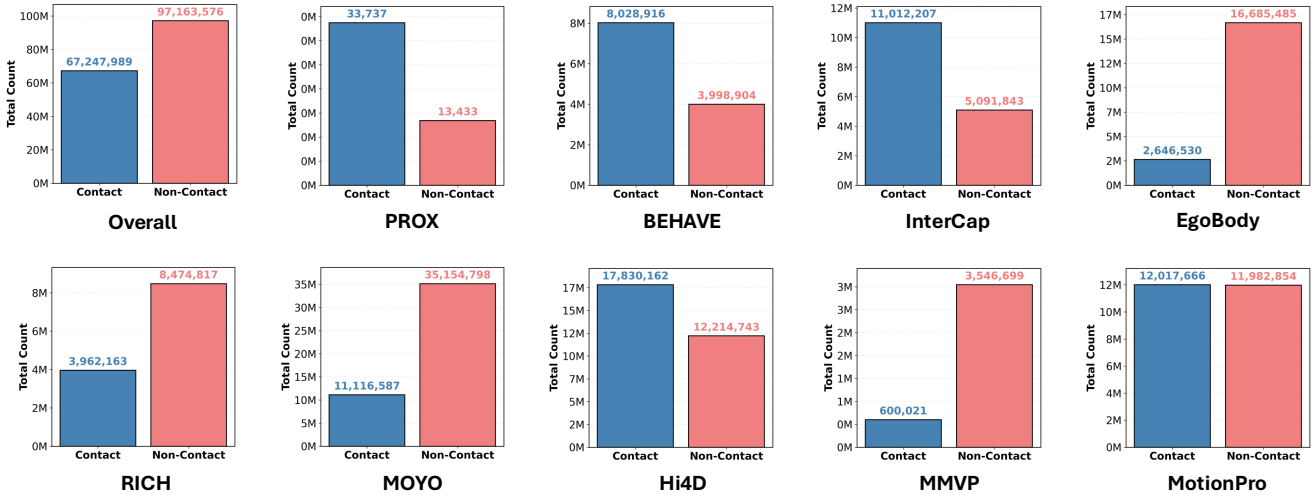

Figure S3. **Contact and non-contact distribution of training datasets for FECO.**

Table S2. **Ground plane configuration.** Negative height refers to datasets whose coordinate system is defined such that smaller values along the height axis correspond to higher positions.

Dataset	Height axis	Ground axis	Negative height
PROX [9]	z-axis	x-axis, y-axis	✗
BEHAVE [2]	y-axis	x-axis, z-axis	✓
InterCap [13]	y-axis	x-axis, z-axis	✓
EgoBody [28]	y-axis	x-axis, z-axis	✗
RICH [12]	y-axis	x-axis, z-axis	✓
MOYO [24]	z-axis	x-axis, y-axis	✗
Hi4D [26]	y-axis	x-axis, z-axis	✗
MMVP [27]	y-axis	x-axis, z-axis	✓
MotionPRO [22]	y-axis	x-axis, z-axis	✗

cantly underrepresented as in Figure S5. By contrast, COFE not only achieves a closer balance between contact and non-contact, but also distributes annotations more evenly across joints. This makes COFE a complementary resource that mitigates the biases present in existing datasets and provides a more stable training signal for joint-level foot contact estimation.

S2. Details of training FECO

The original SagNets [17] adopt three separate optimizers, where one trains the content-biased network with a task loss and two others train the style-biased network with both a task loss and an adversarial loss. While this has a reasonable computational cost for image classification task, this strategy becomes computationally expensive when extended to dense foot contact estimation. To reduce the cost, we train our model in an end-to-end manner with a single optimizer, which allows the backbone network (*e.g.,* ViT, ResNet), the foot contact decoders, the ground-aware decoders, and the style branch to be optimized jointly. This is achieved with gradient detaching and module freezing for irrelevant modules to resemble the training of the original SagNets. Moreover, all of the operations after low-level style randomization are conducted three parallel operation onto the original input and two low-level style randomized images with ProRandConv [4]. This allows FECO to reduce low-level style bias, which improve model's stability and robustness on unseen shoe styles. Below, we further explain in detail on the process and clarify our design choice on training.

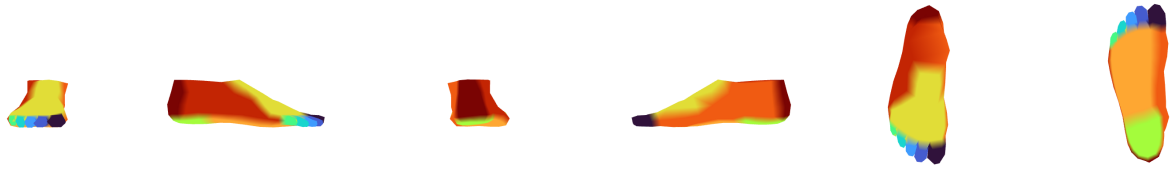

Figure S4. **Foot part segmentation.** We build foot part segmentation that consists of 11 parts. Each part is defined by 5 toes (blue), heel (neon lime green), front (yellow), bottom (bright orange), left (dark orange), right (red), back (dark red).

Gradient detaching for foot segmentation. During the forward pass, the backbone produces multi-level intermediate features $\{\mathbf{F}_l\}_{l=1}^{n-1}$ and a final high-resolution feature map $\mathbf{F}_n$, where there are n layers for the backbone network. From these features $\{\mathbf{F}_l\}_{l=1}^{n}$, a foot segmentation mask $\mathbf{M}_\text{f}$ is predicted and binarized to detect the spatial region of the foot. We intentionally detach the gradient flow from later modules that utilize the foot segmentation mask $\mathbf{M}_\text{f}$, so that the foot segmentation mask decoder is solely trained by the foot segmentation loss $\mathcal{L}_\text{mask}$.

Gradient detaching and module freezing for end-to-end training. There are two major components of training techniques of SagNets that requires modification to enable training of FECO on a single optimizer. First, the style-biased loss of SagNets only train style-biased network., which each corresponds to style loss $\mathcal{L}_\text{style}$ and foot contact decoder in our style branch. This is implemented in SagNets by adopting a separate optimizer that only optimizes the parameters in style-biased network with a dedicated optimizer. However, this can be easily implemented by simply detaching the gradient flow for the input of style-biased network. For FECO, we therefore detach the graident flow of input content invariant feature towards foot contact decoder. This allows our style loss $\mathcal{L}_\text{style}$ to only train foot contact decoder. However, the challenging part is that we need to obtain gradient from the same foot contact decoder in the style path of our FECO to allow adversarial training with our style adversarial loss $\mathcal{L}_\text{style-adv}$. In SagNets, the adversarial loss is applied to the affine parameters of the batch normalization layers within ResNet backbone. Similar with their style loss, they also utilize dedicated optimizer that only optimizes the affine parameters with the adversarial loss. Our FECO instead builds two adapters $\mathbf{A}_\text{prev}$ and $\mathbf{A}_\text{after}$, which each adapts multi-level features $\{\mathbf{F}_l\}_{l=1}^{n}$ and adapts features after content randomization. Nevertheless, there is another challenge. The adapters, which are our replacement of the affine parameters in ResNet backbone from SagNets, are at the initial stages of the FECO model while adversarial loss should only train the adapters $\mathbf{A}_\text{prev}$ and $\mathbf{A}_\text{after}$ without influencing subsequent modules such as foot contact decoder in style branch of FECO. This is difficult as loss inevitably influences all subsequent modules by nature. To tackle this issue, we freeze all parameters within foot contact decoder within style branch of FECO and make only the adapters $\mathbf{A}_\text{prev}$ and $\mathbf{A}_\text{after}$ to be trained with adversarial loss $\mathcal{L}_\text{style-adv}$. This enables us to strictly follow the training process of SagNets while utilizing only a single optimizer, which significantly decreases the computational burden of training.

S3. Details of joint definitions

In Figure S4, we present the foot part segmentation used to extract foot joints for joint level foot contact supervision. This corresponds to $v_2 = 11$. The most widely used foot joint definition of OpenPose [3] provides a foot joint definition that corresponds to $v_3 = 3$, which is highly sparse and captures contact only at the big toe, small toe, and heel, limiting coverage of the remaining foot surface. But, we need a denser joint representation that covers the entire foot surface. Therefore, we start from the SMPL-X [21] foot mesh, which is a subset of the full body model. We partition the foot mesh into eleven intuitive parts that align with functional regions of the foot. First, we segment the five toes, visualized from dark to light blue in Figure S4. Then, we segment the heel to maintain compatibility with the OpenPose joint definition. Furthermore, we segment the remaining plantar surface that frequently contacts the ground. Finally, we divide the dorsal surface into four regions in the left, right, front, and back directions. This yields eleven parts used for both foot part segmentation and the associated joints. Each foot joint is then defined as the mean of the vertex coordinates within its corresponding part.

S4. Details of dense foot contact labels

Following the previous work on hand contact estimation [14], we implement distance-based thresholding with Trimesh library [5] to gather ground-truth dense foot contact labels. However, unlike HACO [14], that had access to mesh of the interacting entity (*i.e.,* 3D object mesh, 3D scene mesh), we only have a few datasets [9, 12, 28] that provide 3D scene mesh. Therefore, in order to also leverage 3D motion capture datasets [2, 13, 22, 24, 26, 27] that do not provide 3D scene mesh, we extract 3D ground mesh and conduct distance-based thresholding between the 3D ground mesh and 3D foot mesh.

Table S3. **Computational requirements of various backbone configurations.**

Model	Backbone	Train Memory (MB)	Test Memory (MB)	Params. (M)	Speed (fps)	GFLOPs
FECO	ViT-H [7]	34,328	6,418	964.64	20.04	190.67
FECO	ViT-L [7]	19,683	4,528	554.37	20.86	73.45
FECO	ViT-B [7]	12,269	3,304	270.44	24.86	24.67
FECO	ViT-S [7]	8,743	2,652	137.38	23.53	6.81
FECO	ResNet-152 [11]	24,410	3,680	335.66	17.98	99.75
FECO	ResNet-101 [11]	21,732	3,644	320.02	20.82	87.49
FECO	ResNet-50 [11]	19,037	3,570	301.03	24.16	72.60
FECO	ResNet-34 [11]	4,757	2,420	104.79	41.07	4.11
FECO	ResNet-18 [11]	4,463	2,378	94.68	42.98	2.26

To extract ground mesh from datasets without 3D scene mesh, we fit a parametric ground plane to each capture sequence. Specifically, we aggregate candidate ground points by collecting the vertices that are lowest in physical height (closest to the real-world ground surface) from both the body and interacting object meshes (only if provided) across frames, and then apply a regression-based plane fitting strategy to obtain coefficients (a, b, c) of the plane $h = ag_1 + bg_2 + c$, where h denotes the height axis (*i.e.*, y-axis of the xyz coordinate system) and g_1, g_2 denote the ground axes (*i.e.*, x-axis and z-axis of the xyz coordinate system).

We also extract ground meshes from datasets with 3D scene mesh. To extract ground mesh from datasets with 3D scene mesh, we directly leverage the 3D scene mesh. We first sample the scene mesh vertices and compute their height values with respect to the vertical axis, taking into account the coordinate convention of each dataset. Among these vertices, we retain only those within the lowest $p\%$ percentile in height, which effectively restricts candidate points to the near-ground region while discarding elevated structures and outliers. We then apply RANSAC-based plane fitting [8] to these candidate points, repeating the procedure multiple times and selecting the plane with the widest spatial support, determined by how many scene mesh vertices fall close to the fitted plane within a distance threshold. This yields the ground plane parameters (a, b, c) of $h = ag_1 + bg_2 + c$, consistent with the coordinate definition used for datasets without 3D scene mesh.

Once the ground plane is estimated, we compute signed distances between the 3D foot mesh vertices and the plane, and assign contact labels when the magnitude of the distance falls within a dataset-specific tolerance. The tolerance values are determined by manually inspecting the dense foot contact results produced under different thresholds and selecting the setting that best aligns both the real-world dense foot contact and consistency between datasets. Specifically, we set the tolerance to 1 cm for MOYO, 2 cm for MotionPRO, 3 cm for PROX and EgoBody, and 5 cm for BEHAVE and InterCap. For datasets that already provide ground-truth dense body contact labels, such as Hi4D and RICH, we directly adopt the provided contact annotations. This unified procedure equips all datasets, regardless of whether they include explicit scene meshes, with dense per-vertex foot contact labels that are geometrically consistent. The ground plane configurations and ground plane parameters used and extracted in this process are each summarized in Table S2 and Table S6.

S5. Visualization of ground-aware learning

Figure S6 presents qualitative results of our ground-aware learning module, which predicts pixel height maps and ground normals from the ground feature. The predicted pixel height maps produced by FECO closely match the ground truth. In particular, the smooth gradient patterns in the pixel height maps indicate that FECO effectively infers per-pixel height relative to the ground for regions corresponding to the foot. Although the predictions exhibit minor smoothing and reduced detail around individual toes, the height maps still provide a strong signal and proof of the ground-aware learning. Furthermore, the predicted ground normals show strong alignment with the corresponding ground-truth normals. The predictions remain robust even when the foot is tilted or not aligned with the ground plane, as illustrated in the third and fourth rows of Figure S6. These accurate ground-aware representations of pixel height and ground normal enable FECO to achieve reliable and precise dense foot contact estimation.

S6. Quantitative results on more datasets

In addition to MMVP [27] dataset, we further validate FECO across diverse datasets to provide additional benchmarks. On BEHAVE [2], FECO achieves strong results on all metrics with an F1-score of 0.795, demonstrating robust performance on foot–object contact estimation. On RICH [12], which contains dense foot–scene interactions, FECO obtains reasonable performance in diverse 3D scene. We also evaluate on MOYO [24], a motion capture dataset designed for extreme yoga poses. Due to the strong out-of-distribution poses and limited scene contact supervision, performance drops compared to other datasets.

Nevertheless, FECO still detects plausible contacts under such challenging conditions. Finally, on Hi4D [26], which includes human–human interactions, our model achieves strong overall results. These findings collectively highlight that FECO not only performs well on MMVP, but also yields reasonable performance across diverse real-world interaction scenarios, including object-centric, extreme-pose, and human interaction settings.

S7. Quantitative results on different backbones

We evaluate the performance of FECO using various backbone architectures on the MMVP [27] dataset, keeping all other components fixed. As summarized in Table S4, the ViT-H [7] backbone achieves the highest F1-score of 0.577, demonstrating the strongest overall performance among all tested models. ViT-based architectures consistently outperform convolutional alternatives, reflecting the benefit of Transformer [25]-based designs in capturing long-range dependencies crucial for dense foot contact estimation. Among the ViT variants, ViT-B attains the highest recall (0.650), suggesting better coverage of subtle contact regions, while ViT-L achieves a balanced trade-off between precision and recall. ViT-S maintains competitive performance despite its compact size, underscoring the scalability of Transformer backbones under resource constraints. Among convolutional backbones, ResNet-152 [11] delivers the strongest result with an F1-score of 0.533, followed by ResNet-101 and ResNet-50. For these models, we employ a decoder based on the re-implementation of FCN [16] from PyTorch [20] to predict pixel height maps. To maintain efficiency, dilation is omitted in shallower variants (ResNet-18 and ResNet-34), though this leads to reduced accuracy due to their limited receptive fields. Overall, these results confirm that ViT backbones are better suited for dense foot contact estimation than convolutional counterparts, primarily due to their global context modeling and superior spatial reasoning. The strong performance of ViT-H further justifies its selection as the default backbone in our main experiments. All model variants will be publicly released.

S8. Computational requirements

Table S3 reports the computational requirements of FECO under different backbone configurations. Large-scale Vision Transformer backbones [7], such as ViT-H, deliver the strongest representational capacity but also incur the highest computational overhead. The ViT-H variant requires more than 34 GB of training memory and nearly 6.5 GB of inference memory. This overhead primarily arises from maintaining strong gradients across the model's 964.64M parameters during training. ViT-L provides a more moderate alternative, reducing both memory usage and computational cost while retaining high model capacity. Smaller variants such as ViT-B and ViT-S provide efficient configurations, with training memory under 13 GB and inference speeds exceeding 23 fps, making them well-suited for practical applications.

ResNet-based backbones [11] further expand the design space by offering a spectrum of efficient choices. ResNet-152 and ResNet-101 provide strong representational power while maintaining stable inference speeds around 20 fps. ResNet-50 achieves a favorable balance, combining manageable memory consumption with fast inference exceeding 24 fps. Lightweight variants such as ResNet-34 and ResNet-18 are highly efficient, requiring fewer than 5 GB of training memory and achieving speeds above 40 fps with very low GFLOPs, making them ideal for real-time scenarios and deployment on resource-constrained hardware.

Overall, FECO supports a wide range of backbones, from high-capacity Transformers to lightweight ResNets, enabling its use across diverse downstream tasks that may have varying computational requirements.

Table S4. **Comparison of various backbone models on MMVP [27] dataset.** All backbones are initialized with ImageNet [6] dataset.

Model	Backbone	Precision↑	Recall↑	F1-Score↑
FECO	ViT-H [7]	<u>0.563</u>	0.613	**0.577**
FECO	ViT-L [7]	**0.578**	0.568	0.567
FECO	ViT-B [7]	0.526	**0.650**	<u>0.574</u>
FECO	ViT-S [7]	0.507	0.620	0.546
FECO	ResNet-152 [11]	0.463	<u>0.647</u>	0.533
FECO	ResNet-101 [11]	0.530	0.550	0.526
FECO	ResNet-50 [11]	0.497	0.579	0.515
FECO	ResNet-34 [11]	0.448	0.550	0.481
FECO	ResNet-18 [11]	0.484	0.458	0.437

Table S5. **Quantitative results on more datasets**

Model	Evaluation Dataset	Precision ↑	Recall ↑	F1-Score ↑
FECO	BEHAVE [2]	0.755	0.917	0.795
FECO	RICH [12]	0.581	0.780	0.621
FECO	MOYO [24]	0.546	0.573	0.504
FECO	Hi4D [26]	0.761	0.888	0.796

S9. More qualitative results

Figure S7 and Figure S8 present additional qualitative comparisons of POSA [10], BSTRO [12], and DECO [23] on the Hi4D [26], MMVP [27], RICH [12], MOYO [24], and COFE dataset. Our FECO consistently outperforms previous state-of-the-art methods by a large margin. DECO frequently predicts full plantar contact regardless of the actual foot contact observable from the image. POSA frequently yields false positives contact prediction when the foot pose

suggests contact despite there is no contact occurrence in the image. For example, in the fifth row of Figure S7, although the foot does not touch the ground, its pose leads POSA to hallucinate non-existent contact. BSTRO fails to infer contact in the heel region. For instance, in the last row of Figure S7 and the second row of Figure S8, BSTRO does not detect heel contact even though it is well present. This behavior arises from the absence of ground-aware learning in BSTRO, whereas FECO explicitly incorporates pixel height maps and ground normals to enable ground-aware contact reasoning. Furthermore, FECO maintains accurate predictions across diverse shoe styles in all samples, which is enabled by its shoe style-invariant learning.

S10. Limitations and societal impacts

Limitations. Our FECO overcomes the challenge of diverse shoe appearance, which is inherent and unique problem for dense foot contact estimation. Also, to overcome the issue of lack of in-the-wild training and evaluation dataset for foot contact, we manually annotate and introduce COFE dataset. However, we mainly operate under foot cropped image, which may not provide useful information when fully occluded. Advancing towards the integration with dense full-body contact estimation methods or proposing additional module that takes information from full image without being biased with full body pose would solve such occlusion problem. Also, dense foot contact can be much easier problem with temporal information as static property of foot during contact is, should not serve as a sole contributor but, important cue for contactness. Developing video-based [18] dense foot contact estimation would greatly improve the performance of the model. Lastly, modeling interactions between the two feet analogous to two hands interaction modeling [19] can be a promising direction for future research.

Societal impacts. The proposed method has broad potential for applications in sports analytics, rehabilitation, AR/VR, and human behavior understanding. Nevertheless, deploying dense foot contact estimation in the wild entails risks related to privacy, safety, and sustainability. Patterns of foot contact during daily activity may reveal aspects of health, so any data collection should occur only with consent, minimal retention, and, when possible, on-device processing. Our method is not designed as a diagnostic tool, and such use would require higher accuracy and fidelity on specific diagnostic scenarios. To build a diagnostic system on top of FECO, one would need calibrated confidence estimates, detection of out-of-distribution inputs, expert oversight, and an intended-use license that restricts surveillance and other misuse.

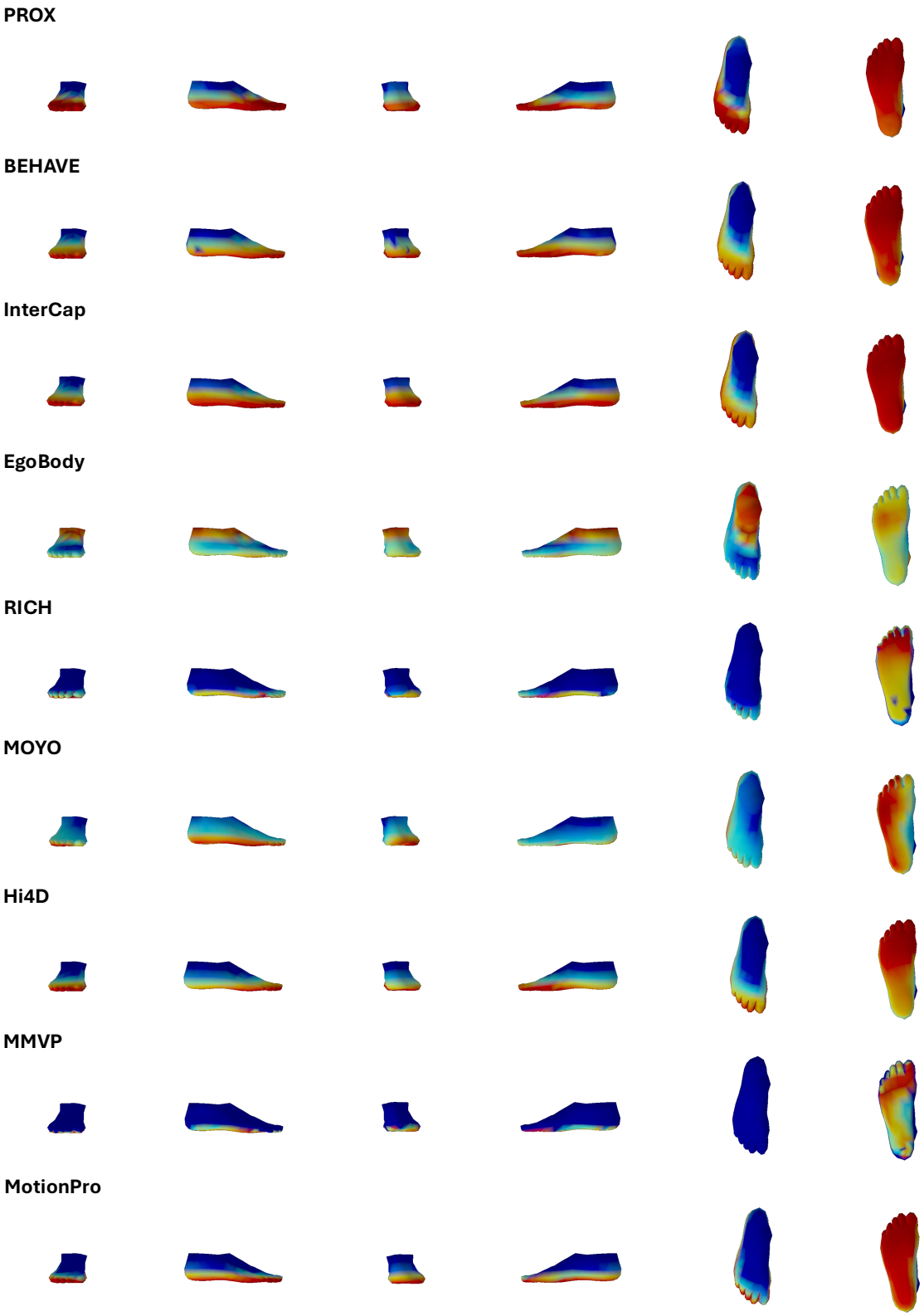

Figure S5. **Dataset-wise dense foot contact mean.** These heatmaps show mean foot contact of ground-truth contacts from PROX [9], BEHAVE [2], InterCap [13], EgoBody [28], RICH [12], MOYO [24], Hi4D [26], MMVP [27], MotionPro [22] dataset.

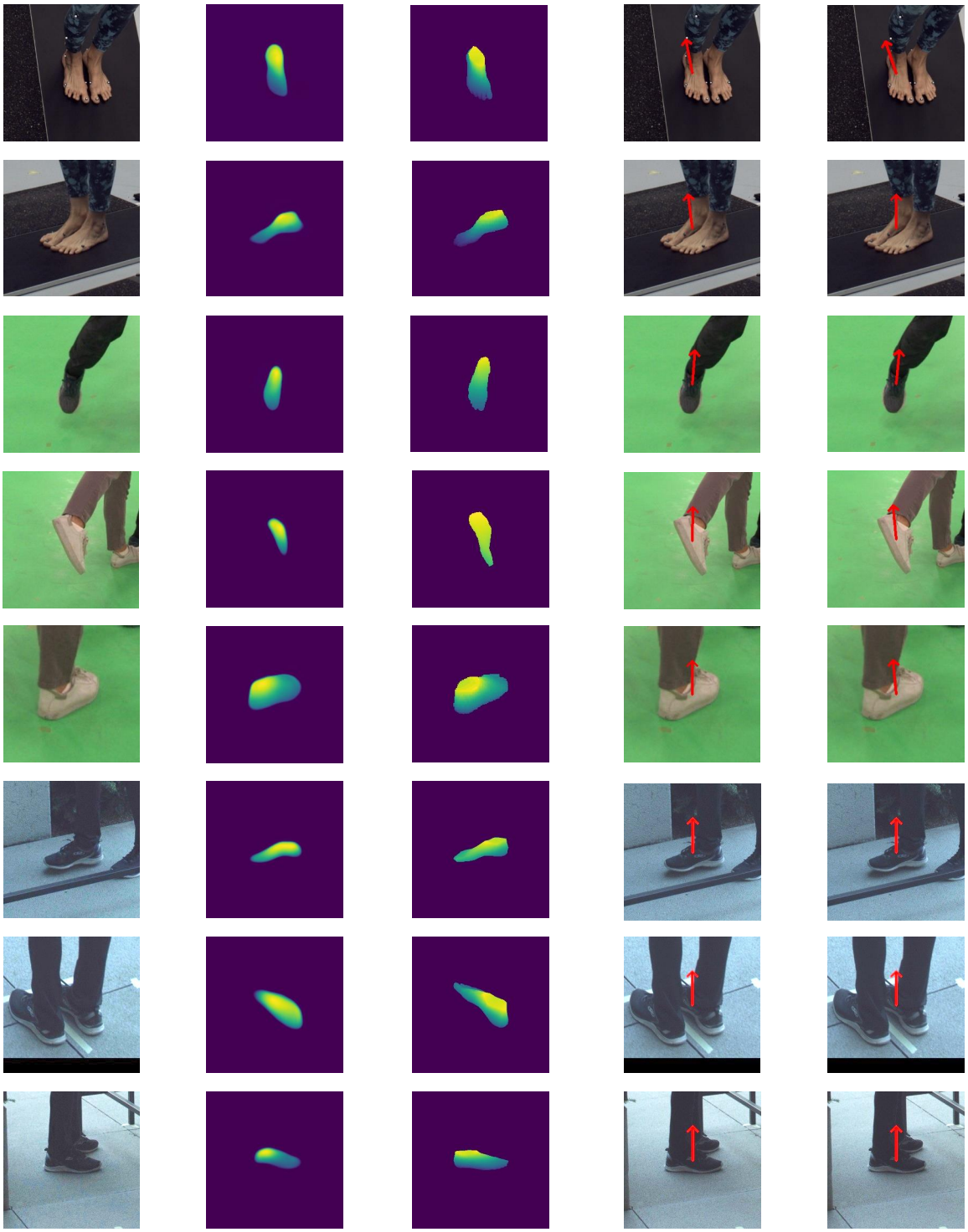

Figure S6. **Visualization of ground-aware learning on MOYO [24], Hi4D [26], RICH [12] dataset.**

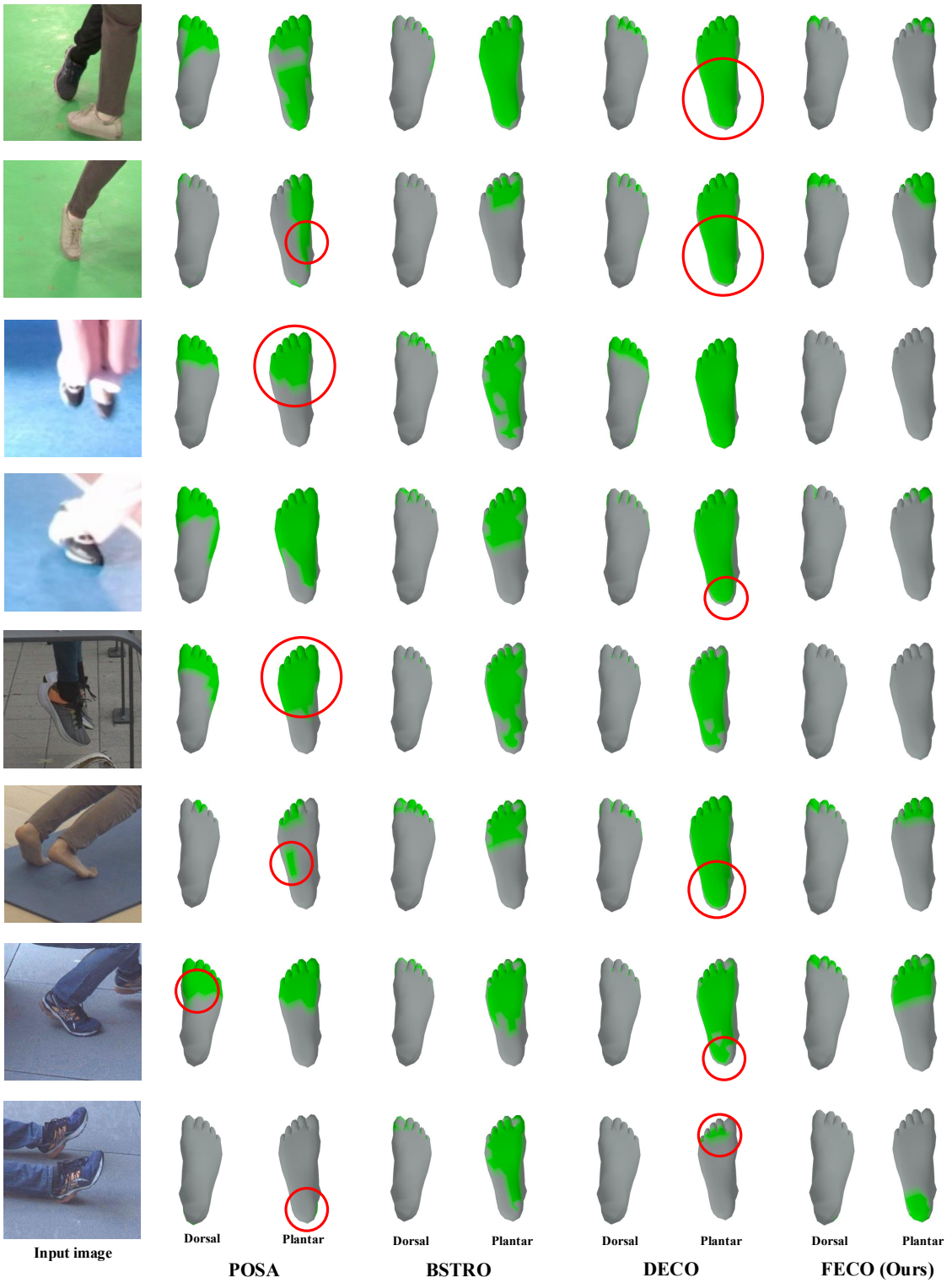

Figure S7. **Qualitative comparison of dense foot contact estimation with POSA [10], BSTRO [12], DECO [23] on Hi4D [26], MMVP [27], RICH [12] dataset.** Red circles indicate exemplar regions that FECO outperforms previous methods.

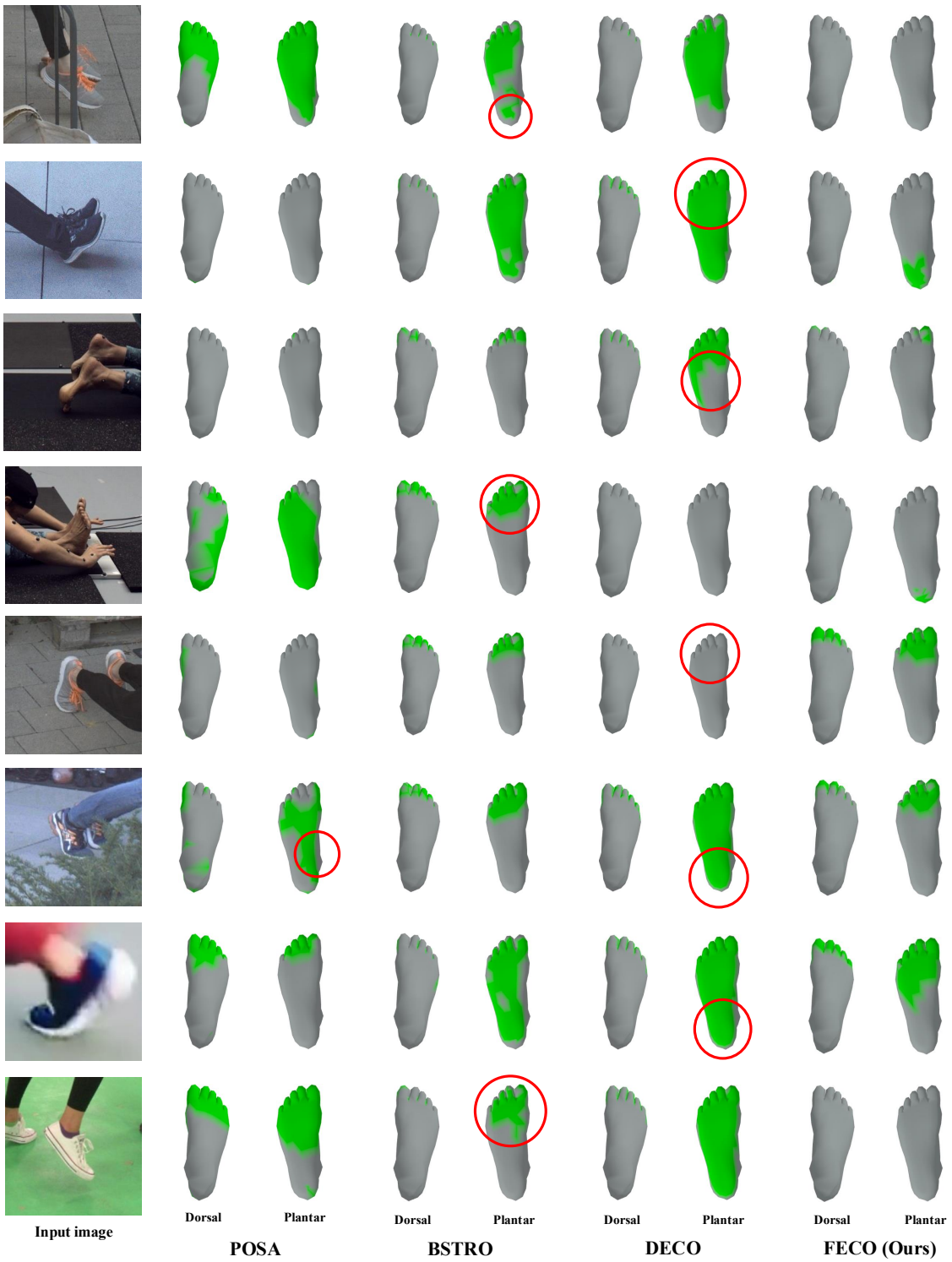

Figure S8. **Qualitative comparison of dense foot contact estimation with POSA [10], BSTRO [12], DECO [23] on RICH [12], MOYO [24], Hi4D [26], COFE dataset.** Red circles indicate exemplar regions that FECO outperforms previous methods.

Table S6. **Coefficients of fitted ground plane of datasets.** All refers to all sequences in the dataset.

Dataset	Sequence	slope of g_1	slope of g_2	intercept of h
PROX [9]	Quantitative	x: 0.005543	y: 0.021068	z: -0.116057
BEHAVE [2]	Date01	x: 0.027887	z: -0.008862	y: 1.201017
	Date02	x: -0.003986	z: 0.010984	y: 1.178358
	Date03	x: 0.033936	z: 0.016988	y: 1.230144
	Date04	x: 0.007792	z: -0.009504	y: 1.194403
	Date05	x: 0.022923	z: -0.012357	y: 1.193551
	Date06	x: 0.019536	z: -0.018724	y: 1.213272
	Date07	x: 0.009183	z: -0.003937	y: 1.217363
InterCap [13]	All	x: 0.027502	z: -0.134394	y: 1.454231
EgoBody [28]	seminar_g110	x: -0.012439	z: 0.002003	y: -1.661606
	seminar_h52	x: 0.001277	z: 0.001124	y: -0.505707
	kitchen_gfloor	x: 0.001684	z: 0.000132	y: -0.842190
	cab_e	x: 0.000703	z: 0.006315	y: -0.711828
	cab_h_tables	x: -0.000753	z: -0.000542	y: -0.331833
	seminar_d78	x: 0.002037	z: 0.000250	y: -0.814285
	seminar_d78_0318	x: -0.001632	z: 0.002296	y: -1.023084
	seminar_g110_0415	x: 0.000735	z: -0.001843	y: -0.789173
	foodlab_0312	x: -0.005830	z: -0.000577	y: -0.709450
	cab_g_benches	x: -0.000604	z: -0.000745	y: -0.155386
	seminar_j716	x: 0.000745	z: -0.001081	y: -0.894595
	seminar_h53_0218	x: -0.000041	z: -0.003415	y: -0.786509
	cnb_dlab_0215	x: 0.000525	z: -0.000740	y: -0.119325
	seminar_g110_0315	x: 0.003623	z: -0.000446	y: -0.733775
	cnb_dlab_0225	x: 0.000715	z: 0.001987	y: -0.155290
RICH [12]	BBQ	x: 0.028968	z: -0.137393	y: 1.421628
	Gym	x: -0.055151	z: -0.237227	y: 1.577146
	LectureHall	x: 0.027296	z: -0.299544	y: 1.979380
	ParkingLot1	x: -0.022767	z: -0.128437	y: 1.372453
	ParkingLot2	x: -0.018011	z: -0.131626	y: 1.375236
MOYO [24]	All	x: 0.000000	z: 0.000000	y: 0.000000
Hi4D [26]	All	x: -0.010825	z: 0.011383	y: 0.274179
MMVP [27]	S01	x: 0.000000	z: 0.000000	y: 1.253302
	S02	x: 0.000000	z: 0.000000	y: 1.245441
	S03	x: 0.000000	z: 0.000000	y: 1.242629
	S04	x: 0.000000	z: 0.000000	y: 1.247209
	S05	x: 0.000000	z: 0.000000	y: 1.219583
	S06	x: 0.000000	z: 0.000000	y: 1.254879
	S07	x: 0.000000	z: 0.000000	y: 1.255879
	S09	x: 0.000000	z: 0.000000	y: 1.240000
	S10	x: 0.000000	z: 0.000000	y: 1.266970
	S11	x: 0.000000	z: 0.000000	y: 1.223758
	S12	x: 0.000000	z: 0.000000	y: 1.246929
MotionPRO [22]	All	x: 0.000000	z: 0.000000	y: 0.000000

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 1

[2] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. BEHAVE: Dataset and method for tracking human object interactions. In *CVPR*, 2022. 2, 3, 4, 5, 7, 11

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. In *TPAMI*, 2019. 1, 3

[4] Seokeon Choi, Debasmit Das, Sungha Choi, Seunghan Yang, Hyunsin Park, and Sungrack Yun. Progressive random convolutions for single domain generalization. In *CVPR*, 2023. 2

[5] Dawson-Haggerty et al. Trimesh. 3

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4, 5

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981. 4

[9] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*, 2019. 2, 3, 7, 11

[10] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. Populating 3D scenes by learning human-scene interaction. In *CVPR*, 2021. 5, 9, 10

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[12] Chun-Hao P Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J Black. Capturing and inferring dense full-body human-scene contact. In *CVPR*, 2022. 2, 3, 4, 5, 7, 8, 9, 10, 11

[13] Yinghao Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. InterCap: Joint markerless 3D tracking of humans and objects in interaction. In *GCPR*, 2022. 2, 3, 7, 11

[14] Daniel Sungho Jung and Kyoung Mu Lee. Learning dense hand contact estimation from imbalanced data. In *NeurIPS*, 2025. 3

[15] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019. 1

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 5

[17] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021. 2

[18] Hyeongjin Nam, Daniel Sungho Jung, Yeonguk Oh, and Kyoung Mu Lee. Cyclic test-time adaptation on monocular video for 3D human mesh reconstruction. In *ICCV*, 2023. 6

[19] JoonKyu Park, Daniel Sungho Jung, Gyeongsik Moon, and Kyoung Mu Lee. Extract-and-adaptation network for 3D interacting hand mesh recovery. In *ICCV*, 2023. 6

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[21] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 3

[22] Shenghao Ren, Yi Lu, Jiayi Huang, Jiayi Zhao, He Zhang, Tao Yu, Qiu Shen, and Xun Cao. MotionPRO: Exploring the role of pressure in human mocap and beyond. In *CVPR*, 2025. 2, 3, 7, 11

[23] Shashank Tripathi, Agniv Chatterjee, Jean-Claude Passy, Hongwei Yi, Dimitrios Tzionas, and Michael J Black. DECO: Dense estimation of 3D human-scene contact in the wild. In *ICCV*, 2023. 5, 9, 10

[24] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3D human pose estimation via intuitive physics. In *CVPR*, 2023. 2, 3, 4, 5, 7, 8, 10, 11

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[26] Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Jie Song, and Otmar Hilliges. Hi4D: 4D instance segmentation of close human interaction. In *CVPR*, 2023. 2, 3, 5, 7, 8, 9, 10, 11

[27] He Zhang, Shenghao Ren, Haolei Yuan, Jianhui Zhao, Fan Li, Shuangpeng Sun, Zhenghao Liang, Tao Yu, Qiu Shen, and Xun Cao. MMVP: A multimodal mocap dataset with vision and pressure sensors. In *CVPR*, 2024. 2, 3, 4, 5, 7, 9, 11

[28] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. EgoBody: Human body shape and motion of interacting people from head-mounted devices. In *ECCV*, 2022. 2, 3, 7, 11

[29] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 1