Supplementary Material:
Joint Hand Motion and Interaction Hotspots Prediction from Egocentric Videos

Shaowei Liu[1*] Subarna Tripathi[2] Somdeb Majumdar[2] Xiaolong Wang[3]
[1]University of Illinois Urbana-Champaign [2] Intel Labs [3]UC San Diego
Project Page: https://stevenlsw.github.io/hoi-forecast/

The supplementary material provides more details, results and visualizations to support the main paper. In summary, we include

- A. Training labels generation details.
- B. Evaluation set annotation details.
- C. Implementation details of training and inference.
- D. Detailed network architecture and corresponding input and output dimensionality.
- E. Additional experiments of comparison Transformer against 3D CNN and adopting end-to-end training.
- F. Automatic generated training labels visualization.
- G. Qualitative comparison with other methods
- H. Cross-environment and cross-dataset generalization results visualization.

A. Training Labels Generation Details

Hand trajectory generation. We provide additional implementation details of future hand trajectory training labels generation. As we project hand locations from all future frames to the last observation frame, we need to handle the case when there are missing hand detections in future frames. We fill the gap of missing time steps by conducting Hermite spline interpolation. Such interpolation guarantees the smoothness and continuousness of the generated trajectory. We generate the future hand trajectory at 20 FPS and sample at 4 FPS for training.

Interaction hotspots generation. For interaction hotspots training label generation, we detect contact points in the contact frame and project them back to the last observation frame by a similar technique as future hand trajectory generation. However, we need to handle the active object case, *i.e.* the object is moved by the hand in future frames, as shown in Figure 1. To this end, we obtain a future active object trajectory similar to the hand and move the contact points to the active object's original place where it stays still in the last observation frame after the projection.

B. Evaluation Annotations Details

We use the Amazon Mechanical Turk platform to collect interaction hotspot annotations on the evaluation set. The interface is shown in Figure 2. We provide the contact frame in the left image and the last observation frame in the right image in the layout. Users are asked to place points on the same object location in the right image touched by the hand in the left image. The green dots are the labeled points by users. We require all the placed points to be visible and touched by the hand in the left image but haven't been touched in the right image. We collect 1-5 contact point labels for each sample in the evaluation set.

Figure 1. Demonstration of how to generate interaction hotspots in active object case. The left image shows the contact frame, while the right image shows the last observation frame. The detected contact points are shown in magenta dots in both frames. We move the detected contact points along the active object future trajectory (yellow line in the left image) to its original place in the last observation frame to compute the correct interaction hotspots.

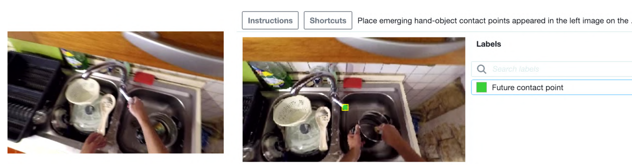

Figure 2. Interface for collecting interaction hotspot annotations on evaluation set. The left image shows the contact frame, while the right image shows the last observation frame. users are asked to place points (green dots) on the same object location in the right image touched by the hand in the left image. All the placed points are visible and on some objects.

C. Implementation Details

In our proposed Transformer model, we set the embedding dimension to be 512 and use a dropout rate of 0.1 for both encoding and decoding blocks. In the C-VAE head network of both hand and object, we implement it as a 2-layer MLP, each for the encoding function $\mathcal{F}_{enc}$ and the decoding function $\mathcal{F}_{dec}$. In the regular training epochs, we use cosine annealed learning

[*]Work partially done during an internship at Intel Labs.

Stage	Configuration	Output
0	Input videos	$T\times256\times454\times3$
	Backbone	
1	TSN [13]	$T\times1024\times8\times14$
1	Hand-RoiAlign [3]	$2\times T\times1024$
1	Object-RoiAlign [3]	$2\times T\times1024$
1	Global-RoiAlign [3]	$T\times1024$
	Hand-Object Detector [11]	
1	Hand location	$2\times T\times4$
1	Object location	$2\times T\times4$
	Pre-processing	
2	Hand MLP	$2\times T\times512$
2	Object MLP	$2\times T\times512$
2	Global MLP	$T\times512$
2	Input tokens	$5\times T\times512$
	OCT Encoder $\mathcal{E}$	
3	encoding blocks $\mathcal{B}$	$5\times T\times512$
	OCT Decoder $\mathcal{D}$	
4	decoding blocks $\mathcal{B}$	$5\times T\times512$
	Hand C-VAE	
5	encoding function $\mathcal{F}_{enc}$	256 (μ)
5	decoding function $\mathcal{F}_{dec}$	2 ($\mathcal{H}$)
	Object C-VAE	
5	encoding function $\mathcal{F}_{enc}$	256 (μ)
5	decoding function $\mathcal{F}_{dec}$	2 ($\mathcal{O}$)

Table 1. Network architecture of the proposed model and corresponding dimensionality.

rate decay starting from $1e-4$. During inference, as we need the hand location in the last observation frame as the 0-th input to the decoder, we set the normalized left hand location to $(0.25,1.5)$ and right hand location to $(0.75,1.5)$ when any of them are invisible, followed [5]. Our model is implemented with PyTorch [9].

Epic-Kitchens. On Epic-Kitchens, our model takes 2.5s observations as input and forecasts future 1s hand trajectory and interaction hotspots. We sample the videos at 4 fps for training and evaluation. We train our model for 35 epochs, including 5 epochs warmup.

EGTEA Gaze+. On EGTEA Gaze+, we set the anticipation time to be 0.5s following [2, 7], given it has a smaller angle of view against the Epic-Kitchens dataset. Our model takes 1.5s observations as input. We sample the videos at 6 fps for training and evaluation. We train our model for 25 epochs, including 5 epochs warmup.

D. Network Architectures

The network architecture is illustrated in Table 1. We utilize ROIAlign [3] to crop the global, hand, and object features in each input frame t with dimension 1024. Then the extracted features and the detected hand and object bounding box locations are fused in the pre-processing module to get Transformer input tokens. The tokens are passed through the OCT encoder and decoder independently. The final future hand trajectory $\mathcal{H}$ at each time step is sampled from the hand C-VAE in an auto-regressive manner. The final object contact points are similarly sampled from the object C-VAE.

Table 2. **Comparison against 3D CNN** on EK100 dataset. (↑/↓ indicates higher/lower is better.) The 3D CNN uses I3D with ResNet-50 as backbone architecture. The proposed transformer model outperforms 3D CNN in both trajectory estimation and interaction hotspots prediction across all metrics.

	Trajectory		Interaction Hotspots		
Model	ADE ↓	FDE ↓	SIM ↑	AUC-J↑	NCC ↑
I3D [1]	0.19	0.16	0.16	0.64	0.55
Ours	**0.12**	**0.11**	**0.19**	**0.69**	**0.72**

Table 3. **Comparison of performance by adopting end-to-end training** of our model on the EK100 dataset. In the paper, we report the performance of utilizing the frozen backbone. Here we compare the performance by training the model end-to-end. We observe a slight performance gain on trajectory estimation. Given that they achieve comparable performance and training end-to-end is more time-consuming, we freeze the backbone in our experiments.

	Trajectory		Interaction Hotspots		
End-to-End	ADE ↓	FDE ↓	SIM ↑	AUC-J↑	NCC ↑
No	0.12	**0.11**	**0.19**	0.69	**0.72**
Yes	**0.11**	**0.11**	**0.19**	**0.70**	0.70

E. Additional Experiments

Comparison to 3D CNNs. We compare our proposed Transformer model with 3D CNN, which is widely used in video understanding. We adopt the I3D [1] with ResNet-50 [4] as backbone for 3D CNN. On the top of the backbone output, we predict the future hand locations and contact points by two head networks, similar to the hand and object head used in OCT. The I3D is pre-trained on Kinetics [6] dataset. We trained the 3D CNN under the same setting as we trained OCT. The performance is shown in Table 2. Experimental results show that by utilizing Transformer architecture against 3D CNN, we could improve the performance on both tasks. The OCT improves the FDE by 58.3% and ADE by 45.5% for trajectory estimation, SIM by +3%, AUC-J by +5%, and NSS by +17% for hotspots prediction on the EK100 dataset. This demonstrates the superiority of adopting Transformer architecture for visual forecasting.

End-to-end training. In the main paper, we freeze the backbone TSN [13] and only train the OCT. We compare the performance against training end-to-end by fine-tuning the backbone along with the OCT. We apply data augmentation including random flipping and color jittering during training. The performance is shown in Table 3. As can be seen, both models achieve comparable performance on both tasks. Given that training end-to-end is more time-consuming, we freeze the backbone in our experiments.

Figure 3. Visualization of the automatically generated training labels on Epic-Kitchens dataset. The right and left future hand trajectory are shown in red and green. The heatmap indicates the interaction hotspots.

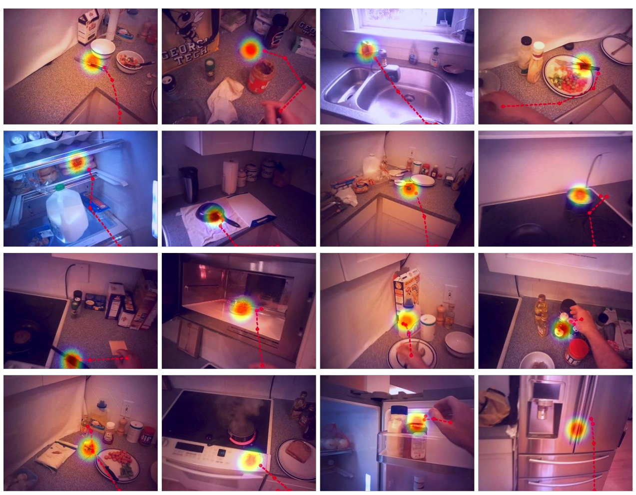

Figure 4. Visualization of the automatically generated training labels on EGTEA Gaze+ dataset. The right and left future hand trajectory are shown in red and green. The heatmap indicates the interaction hotspots.

F. Training Labels Visualization

We visualize the automatically generated training labels on Epic-Kitchens and EGTEA Gaze+ datasets in Figure 3 and Figure 4. It can be seen from the figures that our method could generate high-quality training labels under different kitchen environments and different subjects.

G. Qualitative Comparisons

We compare our model's prediction of future hand trajectory and interaction hotspots against methods that achieved second-best performance in each task, as reported in Table 1 and Table 2 in the main paper.

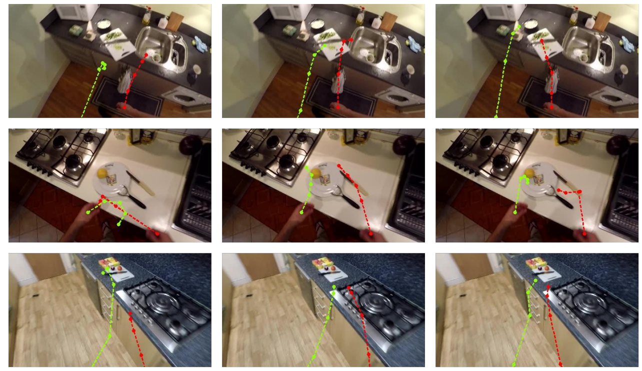

Figure 5. Qualitative comparison of future hand trajectory against **Seq2Seq** [12] on the EK100 dataset. The **first column** shows the **Seq2Seq prediction**, the **second column** shows results of **our method**, and the **third column** are the **ground-truth**. The right and left hand trajectory are shown in red and green. Our method's prediction is more close to ground-truth against Seq2Seq and better reflects human's intention.

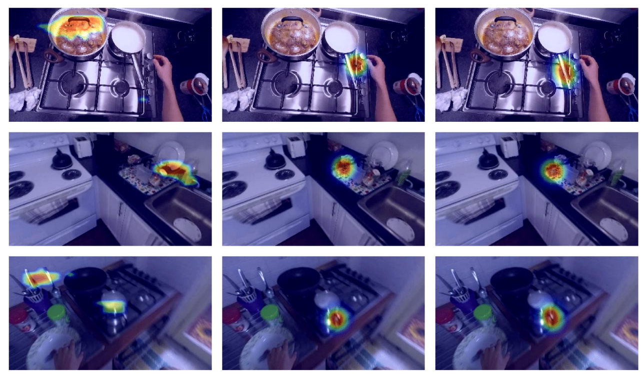

Figure 6. Qualitative comparison of interaction hotspots estimation against **Hotspots** [8] on EK100 dataset. The **first column** shows the **Hotspots prediction**, the **second column** shows results of **our method**, and the **third column** are the **ground-truth**. From the visualization, we observe Hotspots fails when there are multiple candidate objects in the cluttered scene, while our method could better capture the future interactions.

Hand trajectory comparison. We visualize the prediction results on the EK100 of our method and SeqSeq [12] that utilizes LSTM for trajectory estimation. The results are shown in Figure 5. As can be seen, our method's prediction is more close to the ground-truth against Seq2Seq and better reflects human intention.

Object interaction hotspots comparison. We compare our model's prediction of interaction hotspots against Hotspots [8] that employ Grad-Cam [10] to infer future hotspots map. Note that Hotspots takes the ground-truth future action label and last observation frame as input. The results are shown in Figure 6. We observe that Hotspots's prediction is struggling when there are multiple objects present in a cluttered scene. This implies forecasting future interaction hotspots is more challenging than the video affordance grounding task solved by Hotspots. The

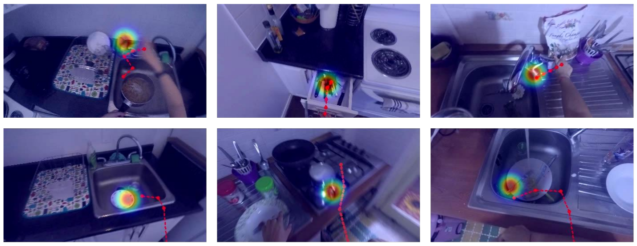

Figure 7. Qualitative visualization of future hand trajectory and interaction hotspots on unseen kitchens and participants on the EK100 dataset. Our model is generalizable to unseen environments and could give reasonable predictions.

Figure 8. Qualitative visualization of cross-dataset generalization results. The model is trained on Epic-Kitchens and tested on EGTEA Gaze+. We show 6 different samples. In each pair of sample, the **left** shows our model prediction, the **right** shows the ground-truth. The future hand trajectory is shown purple. Our model demonstrates strong cross-domain generalization.

future hotspots estimation needs observation frames as context to locate future hand-object interactions.

H. Generalization Results Visualization

Generalization on the unseen kitchens. We visualize our model's prediction on the unseen environment on the EK100 dataset. The selected samples come from the validation split that contains unseen kitchens and participants. We show 6 different samples in Figure 7. Though the kitchen environment is unseen in training, our model could still predict reasonable future hand trajectory and interaction hotspots, which shows the in-domain generalization ability of our model.

Cross-dataset generalization. We visualize the cross-dataset generalization ability on the EGTEA Gaze+ dataset. The model is trained on Epic-Kitchens and tested on EGTEA Gaze+. We show 6 different samples in Figure 8. Our model could well capture the human intention under unseen environments and subjects, forecasting future hand trajectory and interaction hotspots close to the ground-truth. It demonstrates the strong cross-domain generalization ability of our model.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2

[2] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021. 2

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[5] Georgios Kapidis, Ronald Poppe, Elsbeth Van Dam, Lucas Noldus, and Remco Veltkamp. Egocentric hand track and object-based human action recognition. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 922–929. IEEE, 2019. 2

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2

[7] Miao Liu, Siyu Tang, Yin Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European Conference on Computer Vision*, pages 704–721. Springer, 2020. 2

[8] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019. 3

[9] 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. *Advances in neural information processing systems*, pages 8026–8037, 2019. 2

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[11] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9869–9878, 2020. 2

[12] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 3

[13] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2