Deep Future Gaze: Gaze Anticipation on Egocentric Videos Using Adversarial Networks (Supplementary Material)

Mengmi Zhang[1,2], Keng Teck Ma[2], Joo Hwee Lim[2], Qi Zhao[3,1], and Jiashi Feng[1]

mengmi@u.nus.edu, {makt,joohwee}@i2r.a-star.edu.sg, qzhao@cs.umn.edu, elefjia@nus.edu.sg
[1]National University of Singapore, Singapore
[2]Institute for Infocomm Research, A*star, Singapore
[3]University of Minnesota, USA

1. Anatomy of Our Model

We introduce the anatomy of our model for reproducibility in this section. We follow exactly the same convention as Torch. The source code is available at our website[1]. Our framework consists of two networks: **GN** and **D** as shown in Figure S1. In **GN**, it encompasses two modules: **Future Frame Generation Module (G)** and **Temporal Saliency Prediction Module (GP)**. Refer to Table S1 for **G** in **GN**, Table S2 for **GP** in **GN** and Table S3 for **D**.

2. Our OST Dataset

We contribute this new dataset for the object search task. This dataset consists of 57 sequences on search and retrieval tasks performed by 55 subjects. Each video clip lasts for 15 minutes on average with the frame rate 10 fps and frame resolution 480×640. Each subject is asked to search for a list of 22 items and move them to the packing location (dining table). These 22 items are one lanyard, one stress ball, one shampoo, one insect repellent, one raincoat, one file, one thumbdrive, one laptop, one pen, one earpiece, one spoon, one cap, one sunblock, one phone charger, one VGA cable, one 1.5L bottle, one stack of name cards, one calculator, one post-it pad, one bag of granola, one flashlight and one day bag.

The experiment site is a fully furnished and functional model home (in the form of a 2-bedroom apartment) including a master bedroom, children's room, living room, open kitchen, dining area, study room, recreational room, bathroom and exercise area. More examplar images from our OST dataset are presented in Figure S2.

As mentioned in the main text, we only use a subset of frames from our OST dataset (those near the collection table) from all the videos for training and setting. These

[1] https://github.com/Mengmi/deepfuturegaze_gan

Temporal Saliency Prediction Module (GP)
input = torch.Tensor(batchSize, 3, frameSize, img.width, img.height)
nn.VolumetricConvolution(3,128, 3,3,3, 1,1, 1, 1,1,1)
nn.ReLU(true)
nn.VolumetricConvolution(128,256, 4,4,4, 2,2,2, 1,1,1)
nn.ReLU(true)
nn.VolumetricConvolution(256, 256, 3,3,3, 1,1,1, 1,1,1)
nn.ReLU(true)
nn.VolumetricConvolution(256, 256, 3,3,3, 1,1,1, 1,1,1)
nn.ReLU(true)
nn.VolumetricFullConvolution(256,1, 4,4,4, 2,2,2, 1,1,1)
nn.ReLU(true)
nn.Squeeze()
nn.View(batchSize, frameSize, -1)
nn.Transpose(1,3)
nn.SoftMax()
nn.Log()
nn.Transpose(1,3)
nn.View(batchSize, 1, frameSize, img.width,img.height)

Table S2. Architecture of Temporal Saliency Prediction Module (GP) in Generator Network (GN)

Discriminator Network (D)
input = torch.Tensor(batchSize, 3, frameSize, img.width, img.height)
nn.VolumetricConvolution(3,128, 4,4,4, 2,2,2, 1,1,1)
nn.LeakyReLU(0.2, true)
nn.VolumetricConvolution(128,256, 4,4,4, 2,2,2, 1,1,1)
nn.VolumetricBatchNormalization(256,1e-3)
nn.LeakyReLU(0.2, true)
nn.VolumetricConvolution(256,512, 4,4,4, 2,2,2, 1,1,1)
nn.VolumetricBatchNormalization(512,1e-3)
nn.LeakyReLU(0.2, true)
nn.VolumetricConvolution(512,1024, 4,4,4, 2,2,2, 1,1,1)
nn.VolumetricBatchNormalization(1024,1e-3)
nn.LeakyReLU(0.2, true)
nn.VolumetricConvolution(1024,2, 2,4,4, 1,1,1, 0,0,0)
nn.View(2):setNumInputDims(4)

Table S3. Architecture of Discriminator Network (D)

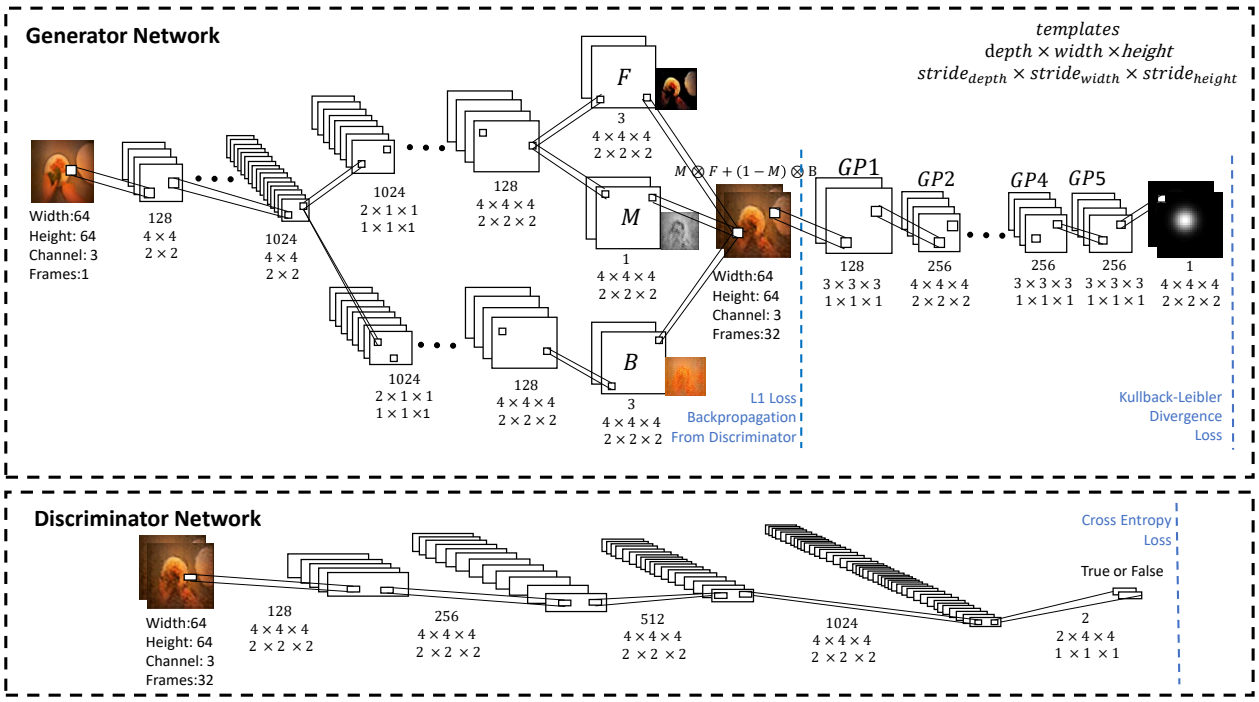


Figure S1. Architecture of Generative Adversarial Network for Gaze Prediction on Current and Future Frames. There are **Generator Network** and **Discriminator Network**. In **Generator Network**, latent representation of the current frame as the input is extracted by 2D convolution layers. It then branches into 3 streams: one for learning the foreground; one for learning the mask to explicitly distinguish foreground and background motions; one for learning the background. These 3 streams are combined to generate future frames. Based on the generated frames, we attach **Temporal Saliency Prediction Module (GP)** to output temporal saliency maps within the next few seconds. The maximum of each predicted temporal saliency map is then the anticipated gaze location. In **Discriminator Network**, it is based upon 3D convolution layers. The output is one label (true or false). All losses are also indicated.

Figure S3. Examplar images from our OST dataset. Row 1 and 2 are from the training set. Row 3 and 4 are from the test set.

frames cover various actions, like taking/putting objects on the table, searching for items on the table and writing, navigating to/away from the table. Examplar frames from our training and testing sets are provided in Figure S3. The full OST dataset is available at our website[1].

3. Statistics of amplitudes for both camera and head motions

G generates future frames. Its two-stream architecture models camera movement by untangling foreground and background motion. **GP** models gaze motion on future frames in the frame coordinate. We provide the statistics of head and gaze motion in our test data in GTEA and GTEAplus datasets. As there is no ground truth for head motion, we estimate it by averaging the dense optical flow in the boundary pixels between adjacent frames. With respect to a frame (480 by 640 in pixels), the statistics of amplitudes for these motion are reported in Table S4. They follow Poisson distribution. The statistics also confirm that both **GP** and **G** are critical for better gaze anticipation by estimating the two motions separately.

4. Effectiveness of GAN on Gaze Anticipation

In our model (DFG), we propose to generate a sequence of future frames using GAN and anticipate gaze on these generated frames. There are several reasons for our design.

Future Frame Generation Module (G)			
input = torch.Tensor(batchSize, 3, img.width, img.height)			
nn.SpatialConvolution(3,128, 4,4, 2,2, 1,1)			
nn.ReLU(true)			
nn.SpatialConvolution(128,256, 4,4, 2,2, 1,1)			
nn.SpatialBatchNormalization(256,1e-3)			
nn.ReLU(true)			
nn.SpatialConvolution(256,512, 4,4, 2,2, 1,1)			
nn.SpatialBatchNormalization(512,1e-3)			
nn.ReLU(true)			
nn.SpatialConvolution(512,1024, 4,4, 2,2, 1,1)			
nn.SpatialBatchNormalization(1024,1e-3)			
nn.ReLU(true)			
nn.View(-1, 1024, 1, 4, 4)			
nn.VolumetricFullConvolution(1024, 1024, 2,1,1)	nn.VolumetricFullConvolution(1024, 1024, 2,1,1)		
nn.VolumetricBatchNormalization(1024)	nn.VolumetricBatchNormalization(1024)		
nn.ReLU(true)	nn.ReLU(true)		
nn.VolumetricFullConvolution(1024, 512, 4,4,4, 2,2,2, 1,1,1)	nn.VolumetricFullConvolution(1024, 512, 4,4,4, 2,2,2, 1,1,1)		
nn.VolumetricBatchNormalization(512)	nn.VolumetricBatchNormalization(512)		
nn.ReLU(true)	nn.ReLU(true)		
nn.VolumetricFullConvolution(512, 256, 4,4,4, 2,2,2, 1,1,1)	nn.VolumetricFullConvolution(512, 256, 4,4,4, 2,2,2, 1,1,1)		
nn.VolumetricBatchNormalization(256)	nn.VolumetricBatchNormalization(256)		
nn.ReLU(true)	nn.ReLU(true)		
nn.VolumetricFullConvolution(256, 128, 4,4,4, 2,2,2, 1,1,1)	nn.VolumetricFullConvolution(256, 128, 4,4,4, 2,2,2, 1,1,1)		
nn.VolumetricBatchNormalization(128)	nn.VolumetricBatchNormalization(128)		
nn.ReLU(true)	nn.ReLU(true)		
nn.VolumetricFullConvolution(128,3, 4,4,4, 2,2,2, 1,1,1)	nn.VolumetricFullConvolution(128,1, 4,4,4, 2,2,2, 1,1,1)		nn.VolumetricFullConvolution(128,3, 4,4,4, 2,2,2, 1,1,1)
nn.Tanh()	nn.Sigmoid()		nn.Tanh()
–	nn.Squeeze()		–
–	nn.MulConstant(-1)	nn.Replicate(3, 2)	–
–	nn.AddConstant(1)	–	–
–	nn.Replicate(3, 2)	–	–
nn.CMulTable()		nn.CMulTable()	
nn.CAddTable()			

Table S1. Architecture of Future Frame Generation Module (G) in Generator Network (GN)

First, compared with a 3D-ConvNet directly modeling gaze anticipation, Frame Generation Module (G) learns the motion infor across both spatial and temporal domains with the additional supervision from the discriminator. The learnt motion cues, which make the generated frames more realistic, are necessary for Temporal Saliency Prediction Module (GP). For validation, we did the ablation study (SalDirect) by removing **GP**: given the current frame at time t, we use a 2D-ConvNet to extract its hidden representation, attach a 3D-ConvNet to predict temporal saliency maps directly, and train in KLD loss.

Results in Figure S4 show DFG outperforms SalDirect in both AAE and AUC. It suggests GAN has essential contributions to gaze anticipation. Moreover, we develop a new model (SalFusion) which averages the temporal saliency maps from both SalDirect and DFG to generate the final temporal saliency maps. SalFusion outperforms two composite models which confirms that the learnt motion cue from GANs is important and complementary to the cues learned directly from SalDirect.

Second, we observe the gaze movement on individual frames is dependent on their previous states; *e.g.* to anticipate gaze on the frame $t+32$, we need to consider gaze transitions across frames by also anticipating gaze on frames t to $t+31$. For verification, we created one baseline: train SALICON model, a 2D-ConvNet, directly for gaze anticipation at time $t+16$ and $t+32$ using their respective ground truth at time $t+16$ and $t+32$. See Table S5 for results. DFG performs much better than SALICON. This suggests the temporal dependence across frames plays fundamental roles in gaze anticipation in egocentric videos and future frame generation using GANs is useful.

	Gaze Motion			Camera Motion		
	Mean	Median	Variance	Mean	Median	Variance
GTEA	20.4	13.5	508	6.7	3.6	92
GTEAplus	7.1	5.0	89	9.9	5.8	135

Table S4. Statistics of camera and gaze motions in GTEA and GTEAplus. All units are in pixels.

5. Study about the effect of the number of frames on gaze anticipation

In video analysis, the number of consecutive frames is a key parameter in practice. To study the effect of the number of frames on which we anticipate gaze, we assign the scalar weights to tune the losses in both **G** and **GP** for the next 32 frames while maintaining the same architecture. For example, we design the weight matrix to be $[1, 1, 1, 1, 0, ..., 0]$ for gaze anticipation in the next 4 frames while ignoring the subsequent frames. In Table S6, we present the averaged

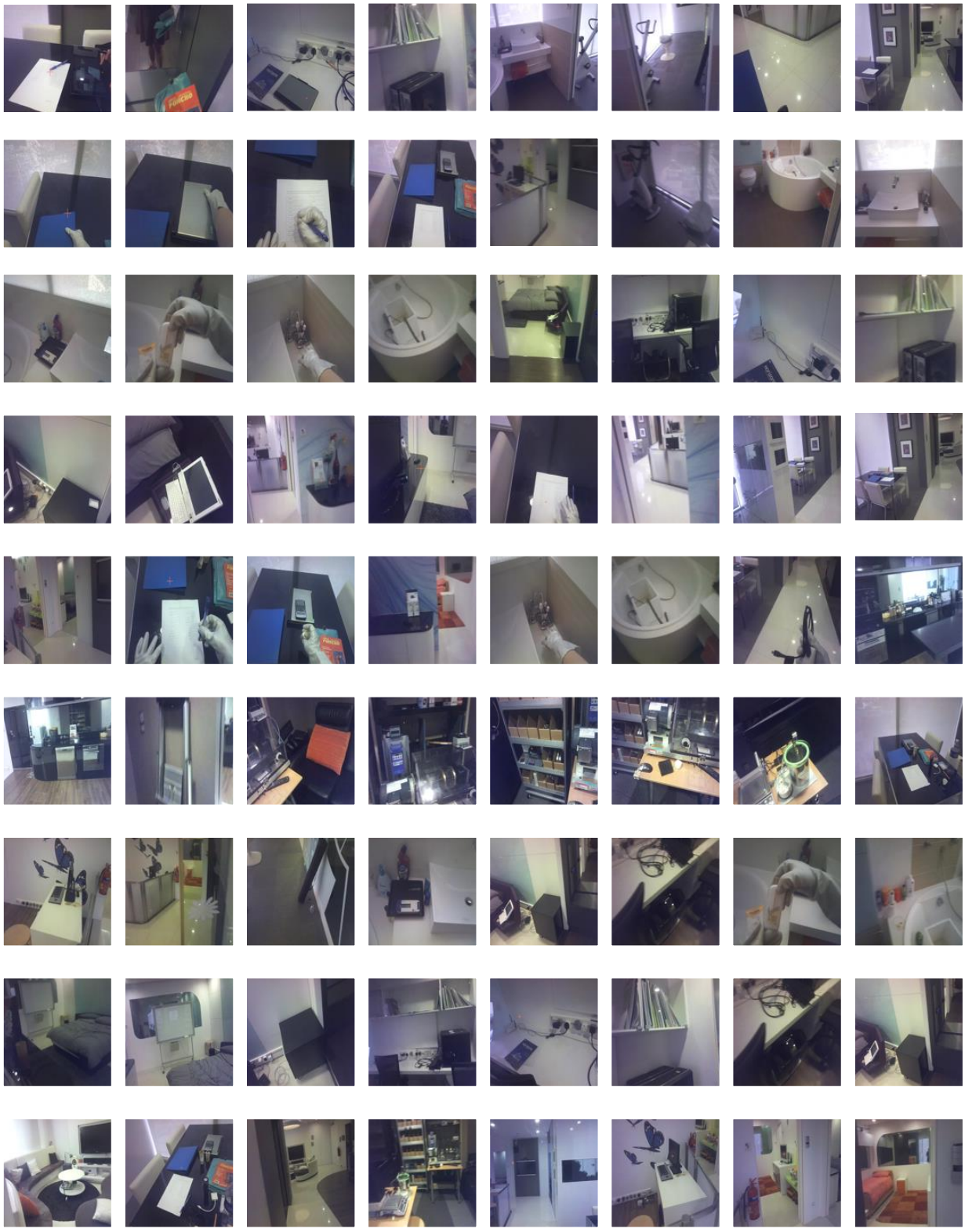
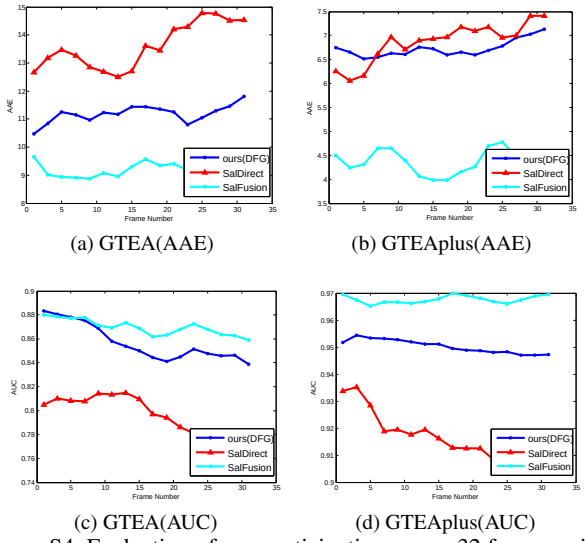


Figure S2. Examplar images from our OST dataset

metric scores of our model for gaze anticipation in the next 2, 4, 8, 16, 32 frames starting from the current frame #1. Detailed discussion about the results is given in the main text.

6. Implementation of Visualization

[7] proposed a top 4 patch visualization approach in 2D-CNN. We extend their work to visualization of 3D-CNN. As a simplified version of their method, we parse all video frames from the test set in GTEA and record the regions with the highest filter activation in both spatial and temporal dimensions for the first and the second last convolution layer in **Temporal Saliency Prediction Module** in our model. Those regions are then projected back into their input video frames based on their corresponding receptive fields across both space and time dimensions where the input frames are the current frame and its subsequent 31 frames. Due to the consistency of egocentric videos between adjacent frames, we increase the diversity of the visualization by sorting the filter activation from highest to lowest and selecting these top filters where their receptive fields do not overlap with their neighboring frames by a pre-defined threshold.



(a) GTEA(AAE) (b) GTEAplus(AAE)

(c) GTEA(AUC) (d) GTEAplus(AUC)

Figure S4. Evaluation of gaze anticipation across 32 frames using AUC and AAE in GTEA and GTEAplus. Ablated models are introduced in Section 4. The higher the better for AUC. The lower, the better for AAE. Best view in color.

Average Angular Error (AAE)				
	GTEAplus		GTEA	
Models	Ours(DFG)	SALICON	Ours(DFG)	SALICON
time $t+16$	**6.6**	11.4	**11.7**	18.4
time $t+32$	**7.5**	19.5	**11.8**	16.6
Area Under Curve (AUC)				
	GTEAplus		GTEA	
Models	Ours(DFG)	SALICON	Ours(DFG)	SALICON
time $t+16$	**0.945**	0.916	**0.850**	0.710
time $t+32$	**0.943**	0.722	**0.838**	0.767

Table S5. Evaluation of gaze anticipation on frames at time $t+16$ and $t+32$ using AUC and AAE on GTEA and GTEAplus. Number denoted in bold is the best.

Angular Average Error (AAE)					
	# 1−2	# 3−4	# 5−8	# 9−16	# 17−32
#2	11.6	–	–	–	–
#4	12.0	12.1	–	–	–
#8	11.4	11.5	11.6	–	–
#16	11.3	10.9	11.3	12.2	–
#32	10.7	11.0	11.2	11.3	11.4
Area Under the Curve (AUC)					
	# 1−2	# 3−4	# 5−8	# 9−16	# 17−32
#2	0.85	–	–	–	–
#4	0.84	0.84	–	–	–
#8	0.86	0.86	0.85	–	–
#16	0.86	0.86	0.86	0.84	–
#32	0.88	0.88	0.88	0.86	0.85

Table S6. Study of correlation between number of frames used for gaze anticipation and corresponding performance of our model. Scores for gaze anticipation in both AAE and AUC are computed every # frames indicated in columns in the testset in GTEA Dataset.

7. Gaze-aided Egocentric Activity Recognition

Recent papers have shown that visual attention could help in egocentric activity recognition [4, 3]. To verify our proposed future gaze model is also useful for egocentric activity recognition, we integrate gaze information into the feedforward 3D-CNN for egocentric activity recognition. As [6] shows that 3D-CNN can be used for activity recognition, we adapt the downscaled framework from [6] (C3D) and integrate the anticipated gaze into the network. A Gaussian mask at the gaze location for each frame, as an additional channel, is concatenated with the input frames of RGB color channels. Cross entropy loss is used for training. Since GTEAplus dataset contains rich instances per activity class as recommended by [4], we follow their evaluation settings and select the top 44 activity classes which have the most instances per class in our recognition task. Confusion matrix of the model with our anticipated gaze is shown in Figure S5. In comparison, we also use the same architecture, discard the gaze information and train the network from scratch. In addition, we provide the baseline that the same architecture with the ground truth gaze information as the upper bound. Since center bias is also effective in gaze prediction, we create an artificial baseline where the network with the center gaze is also evaluated. Activity recognition rates are reported in Table S7.

From Table S7, one can observe our gaze-aided model surpasses C3D network [6] and several traditional methods, STIP [2], Cuboids [1] and guess at random significantly. By comparing the model with our predicted gaze and the one with the center gaze, it can be found that more accurate gaze prediction could result in better egocentric activity

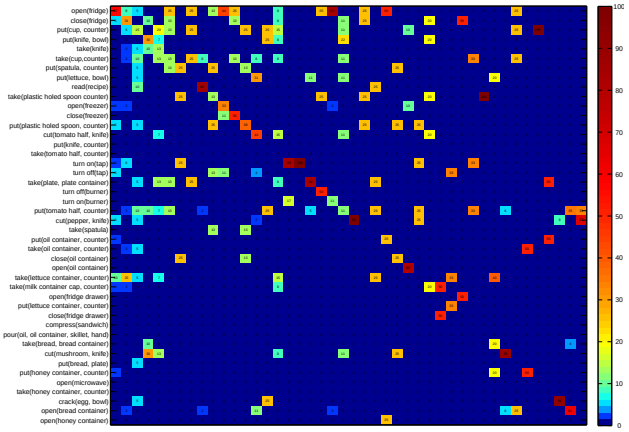


Figure S5. Confusion matrix of 44 egocentric activity classes from GTEAplus Dataset. The 44 activity classes are selected similar as [4, 5]. The results are based upon C3D convolution architecture proposed by [6] for egocentric activity recognition with the fusion of our predicted gaze locations via one convolution layer.

Models	Activity Recognition Rate
Guess At Random	2.3%
STIP	14.9%
CUboids	22.7%
C3D	26.9%
C3D + center gaze	13.6%
C3D + our pred gaze	28.5%
C3D + ground truth gaze	33.5%

Table S7. Accuracy of the Gaze-aided Egocentric Activity Recognition in GTEAplus Dataset.

recognition. However, the wrong gaze information may be misleading for the network, which may result in poor performances as the baseline uses the center bias.

References

[1] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005. 5

[2] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 5

[3] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3216–3223, 2013. 5

[4] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015. 5, 6

[5] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. *arXiv preprint arXiv:1605.03688*, 2016. 6

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014. 5, 6

[7] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 5