In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

EGO-TOPO: Environment Affordances from Egocentric Video (Supplementary Material)

Tushar Nagarajan[1] Yanghao Li[2] Christoph Feichtenhofer[2] Kristen Grauman[1,2]
[1] UT Austin [2] Facebook AI Research
tushar@cs.utexas.edu, {lyttonhao, feichtenhofer, grauman}@fb.com

This document contains supplementary material to support the main paper text. The contents include:

- (§S1) A video demonstrating our EGO-TOPO graph construction process following Algorithm 1, and our scene affordance results from Sec. 4.1 in the main paper.
- (§S2) Setup and details for crowdsourced affordance annotation on EPIC and EGTEA+.
- (§S3) Class-level breakdown of affordance prediction results from Table 1.
- (§S4) Additional implementation details for the graph construction in Sec. 3.1.
- (§S5) Implementation details for our models presented in Sec. 3.3 and Sec. 3.4.
- (§S6) Implementation details for ACTIONMAPS baseline from Sec. 4.1 (Baselines).
- (§S7) Implementation details for SLAM from Sec. 4.1 (Baselines).
- (§S8) Additional affordance prediction results to supplement Fig. 6.

S1. EGO-TOPO demonstration video

We show examples of our graph construction process over time from egocentric videos following Algorithm 1 in the main paper. The end result is a topological map of the environment where nodes represent primary spatial zones of interaction, and edges represent commonly traversed paths between them. Further, the video demonstrates our affordance prediction results from Sec. 4.1 over the constructed topological graph. The video and interface to explore the topological graphs can be found on the project page.

Fig. S2 shows static examples of fully constructed topological maps from a single egocentric video from the test sets of EPIC and EGTEA+. Graphs built from long videos with repeated visits to nodes (P01_18, P22_07) result in a more complete picture of the environment. Short videos where only a few zones are visited (P31_14) can be linked to other graphs of the same kitchen (Sec. 3.2). The last panel shows a result on EGTEA+.

S2. Crowdsourced affordance annotations

As mentioned in Sec. 4.1, we collect annotations for afforded interactions for EPIC and EGTEA+ video frames to evaluate our affordance learning methods. We present annotators with a single frame (center frame) from a video clip and ask them to select *all likely* interactions that occur in the location presented in the clip. Note that these annotations are used exclusively for evaluating affordance models — they are trained using single-clip interaction labels (See Sec. 3.3).

On EPIC, we select 120 interactions (verb-noun pairs) over the 15 most frequent verbs and for common objects that afford multiple interactions. For EGTEA+, we select all 75 interactions provided by the dataset. A list of all these interactions is in Table S1. Each image is labeled by 5 distinct annotators, and only labels that 3 or more annotators agree on are retained. This results in 1,020 images for EGTEA+ and 1,155 images for EPIC. Our annotation interface is shown in Fig. S1 (top panel), and examples of resulting annotations are shown in Fig. S1 (bottom panel).

S3. Average precision per class for affordances

As noted in our experiments in Sec. 4.1, our method performs better on low-shot classes. Fig. S3 shows a class-wise breakdown of improvements achieved by our model over the CLIPACTION model on the scene affordance task. Among the interactions, those involving objects that are typically tied to a single physical location, highlighted in red (*e.g.*, fridges, stoves, taps etc.), are easy to predict, and do not improve much. Our method works especially well for interaction classes that occur in multiple locations (*e.g.*, put/take spoons/butter, pour rice/egg etc.), which are linked in our topological graph.

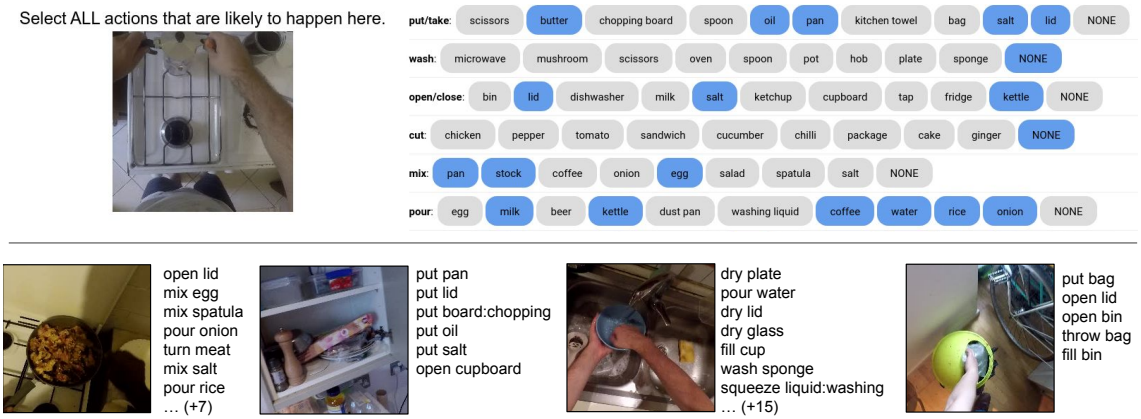

Figure S1: Crowdsourcing affordance annotations. (Top panel) Affordance annotation interface. Users are asked to identify all *likely* interactions at the given location. 6 out of 15 afforded actions are shown here. **(Bottom panel)** Example affordance annotations by Mechanical Turk annotators. Only annotations where 3+ workers agree are retained.

EPIC	put/take: pan, spoon, lid, board:chopping, bag, oil, salt, towel:kitchen, scissors, butter; open/close: tap, cupboard, fridge, lid, bin, salt, kettle, milk, dishwasher, ketchup; wash: plate, spoon, pot, sponge, hob, microwave, oven, scissors, mushroom; cut: tomato, pepper, chicken, package, cucumber, chilli, ginger, sandwich, cake; mix: pan, onion, spatula, salt, egg, salad, coffee, stock; pour: pan:dust, onion, water, kettle, milk, rice, egg, coffee, liquid:washing, beer; throw: onion, bag, bottle, tomato, box, coffee, towel:kitchen, paper, napkin; dry: pan, plate, knife, lid, glass, fork, container, hob, maker:coffee; turn-on/off: kettle, oven, machine:washing, light, maker:coffee, processor:food, switch, candle; turn: pan, meat, kettle, hob, filter, sausage; shake: pan, hand, pot, glass, bag, filter, jar, towel; peel: lid, potato, carrot, peach, avocado, melon; squeeze: sponge, tomato, liquid:washing, lemon, lime, cream; press: bottle, garlic, dough, switch, button; fill: pan, glass, cup, bin, bottle, kettle, squash
EGTEA+	inspect/read: recipe; open: fridge, cabinet, condiment_container, drawer, fridge_drawer, bread_container, dishwasher, cheese_container, oil_container; cut: tomato, cucumber, carrot, onion, bell_pepper, lettuce, olive; turn-on: faucet; put: eating_utensil, tomato, condiment_container, cucumber, onion, plate, bowl, trash, bell_pepper, cooking_utensil, paper_towel, bread, pan, lettuce, pot, seasoning_container, cup, bread_container, cutting_board, sponge, cheese_container, oil_container, tomato_container, cheese, pasta_container, grocery_bag, egg; operate: stove, microwave; move-around: eating_utensil, bowl, bacon, pan, patty, pot; wash: eating_utensil, bowl, pan, pot, hand, cutting_board, strainer; spread: condiment; divide/pull-apart: onion, paper_towel, lettuce; clean/wipe: counter; mix: mixture, pasta, egg; pour: condiment, oil, seasoning, water; compress: sandwich; crack: egg; squeeze: washing_liquid

Table S1: List of afforded interactions annotated for EPIC and EGTEA+.

S4. Additional implementation details for EGO-TOPO graph creation

We provide additional implementation details for our topological graph construction procedure from Sec. 3.1 and Sec. 3.2 in the main paper.

Homography estimation details (Sec. 3.1). We generate SuperPoint keypoints [9] using the pretrained model provided by the authors. For each pair of frames, we calculate the homography using 4 random points, and use RANSAC to maximize the number of inliers. We use inlier count as a measure of similarity.

Similarity threshold and margin values in Algorithm 1 (σ, m). We fix our similarity threshold $\sigma = 0.7$ to ensure that only highly confident views are included in the graph. We select a large margin $m = 0.3$ to make sure that irrelevant views are readily ignored.

Node linking details (Sec. 3.2). We use hierarchical agglomerative clustering to link nodes across different environments based on functional similarity. We set the similarity threshold below which nodes will not be linked as 40% of the average pairwise similarity between every node. We found that threshold values around this range (40-60%) produced a similar number of clusters, while values beyond them resulted in too few nodes linked, or all nodes collapsing to a single node.

Other details. We subsample all videos to 6 fps. To calculate $s_f(f_t, n)$ in Equation 2, we average scores for a window of 9 frames around the current frame, and we uniformly sample a set of 20 frames for each visit for robust score estimates.

S5. Training details for affordance and long term anticipation experiments

We next provide additional implementation and training details for our experiments in Sec. 4 of the main paper.

Affordance learning experiments in Sec. 4.1. For all models, we use ImageNet pretrained ResNet-152 features for frame feature inputs. As mentioned in Sec. 3.3, we use binary cross entropy (BCE) for our loss function. For original clips labeled with a single action label, we evaluate BCE for only the positive class, and mask out the loss contributions for all other classes. Adam with learning rate 1e-4, weight decay 1e-6, and batch size 256 is used to optimize the models parameters. All models are trained for 20 epochs, and learning rate is annealed once to 1e-5 after 15 epochs.

Long term action anticipation experiments in Sec. 4.2. We pretrain an I3D model with ResNet-50 as the backbone on the original clip-level action recognition task for both EPIC-Kitchen and EGTEA+. Then, we extract the features from the pretrained I3D model for each set of 64 frames as the clip-level features. These features are used for all models in our long-term anticipation experiments.

Among the baselines, we implement TRAINDIST, I3D, RNN, and ACTIONVLAD. For TIMECEPTION, we import

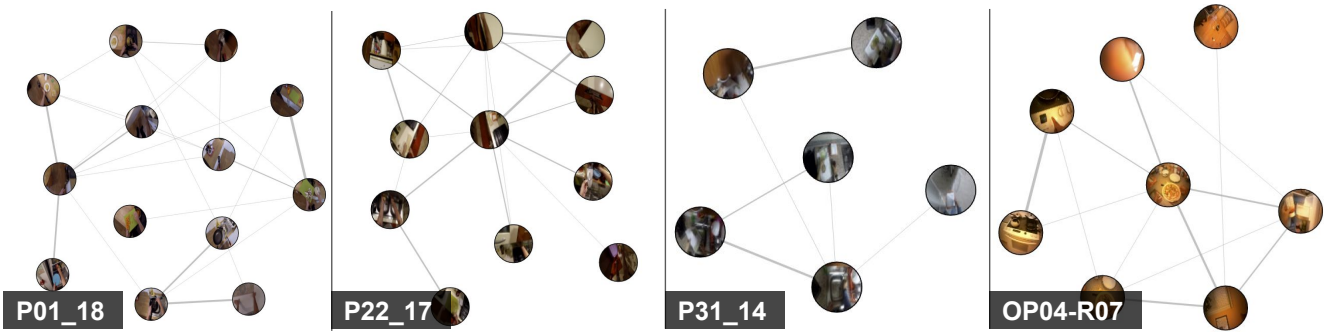

Figure S2: EGO-TOPO graphs constructed directly from egocentric video. Each panel shows the output of Algorithm 1 for videos in EPIC (panels 1-3) and EGTEA+ (panel 4). Connectivity represents frequent paths taken by humans while using the environment. Edge thickness represents how frequently they are traversed.

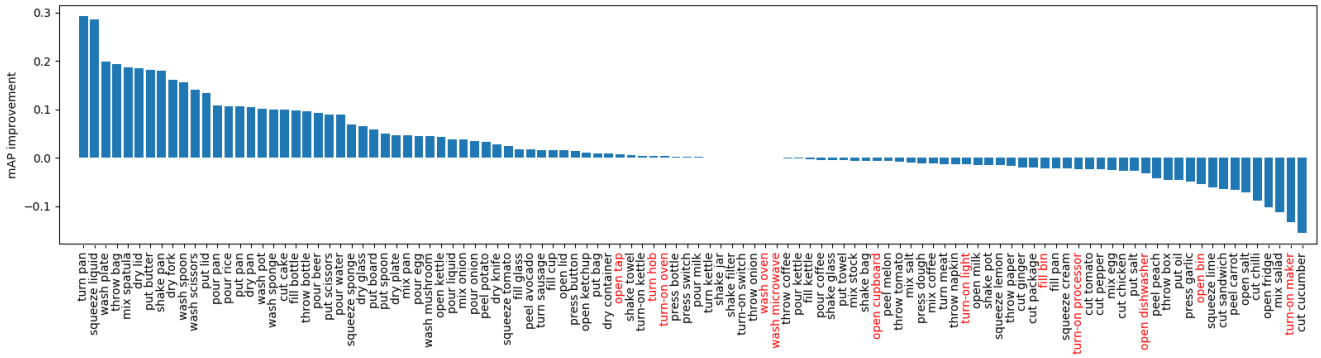

Figure S3: Class-wise breakdown of average precision for affordance prediction on EPIC. Our method outperforms the CLIPACTION baseline on the majority of classes. Single clip labels are sufficient for interactions that are strongly tied to a single physical location (red), whereas our method works particularly well for classes with objects that can be interacted with at multiple locations.

the authors' module[1] and for VIDEOGRAPH, we directly use the authors' implementation[2] with our features as input.

For EPIC, all models are trained for 100 epochs with the learning rate starting from 1e-3 and decreased by a factor of 0.1 after 80 epochs. We use Adam as the optimization method with weight decay 1e-5 and batch size 256. For the smaller EGTEA+ dataset, we follow the same settings, except we train for 50 epochs.

S6. ACTIONMAPS implementation details

For the ACTIONMAPS method, we follow Rhinehart and Kitani [57] making a few necessary modifications for our setting. We use cosine similarity between pretrained ResNet-152 features to measure semantic similarity between locations as side information, instead of object and scene classifier scores, to be consistent with the other evaluated methods. We use the latent dimension 256 for the matrix factorization, and set $\lambda = \mu = 1e-3$ for the RWNMF optimization objective in [57]. We use location information in the similarity kernel only when it is available, falling back to just feature similarity when it is not (due to SLAM failures). We use this baseline in our experiments in Sec. 4.1.

S7. SLAM implementation details

We generate monocular SLAM trajectories for egocentric videos using the code and protocol from [20]. Specifically, we use ORB-SLAM2 [51] to extract trajectories for the full video, and drop timesteps where either tracking is unreliable or lost. We scale all trajectories by the maximum movement distance for each kitchen, so that (x, y) coordinates are bounded between [0, 1]. We create a uniform grid of squares, each with edge length 0.2 . We use this grid to accumulate trajectories for the SLAM baseline and to construct the ACTIONMAPS matrix in our experiments in Sec. 4.1. We use the same process for EPIC and EGTEA+, with camera parameters from the dataset authors.

We experimented with varying grid cell sizes (10x10, 20x20 grids), however, smaller grid cells resulted in very few trajectories registered to the same grid cell (*e.g.*, for a 20x20 grid on EPIC, 61% of cells register only a single trajectory) limiting the amount of labels that can be shared, and hence weakening the baseline. See Table S2.

[1] https://github.com/noureldien/timeception
[2] https://github.com/noureldien/videograph

	EPIC (mAP)	EGTEA+ (mAP)
SLAM_5	41.8	26.5
SLAM_{10}	41.3	26.5
SLAM_{20}	40.7	26.2

Table S2: Affordance prediction results with varying grid sizes. SLAM_S refers to the SLAM baseline from Sec. 4.1 with an $S \times S$ grid.

S8. Additional node affordance results

Fig. S4 provides more examples of affordance predictions by our model on zones (nodes) in our topological map, to supplement Fig. 6 in the main paper. For clarity, we show 8 interactions on EPIC (top panel) and EGTEA+ (bottom panel), out of a total of 120 and 75 interactions respectively.

Figure S4: Additional zone affordance prediction results. Results on EPIC (top panel) and EGTEA+ (bottom panel).